(19) KOREAN INTELLECTUAL PROPERTY OFFICE

KOREAN PATENT ABSTRACTS

(11)Publication number: 1020020020919 A
(43)Date of publication of application: 16.03.2002

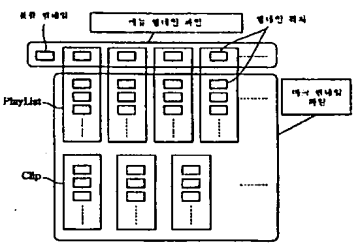
(21)Application number: 1020017016423
(22)Date of filing: 21.12.2001
(30)Priority: JP2000 2000183770 21.04.2000
(51)Int. Cl G11B 20/10

(71)Applicant: SONY CORPORATION
(72)Inventor: KATO MOTOKI
HAMADA TOSHIYA

(54) INFORMATION PROCESSING APPARATUS AND METHOD, PROGRAM, AND RECORDED MEDIUM

(57) Abstract:

A ClipMark composed of a mark indicating a characteristic image extracted from an inputted AV stream is created as management information for managing the AV stream. A PlayListMark composed of a mark indicating an image arbitrarily specified by the user is created from a reproduction section corresponding to a PlayList defining a combination of predetermined sections in the AV stream. The ClipMark and PlayListMark are recorded as mutually independent tables on a recording medium. Therefore a quick and reliable access to a desired position of the AV stream can be made.



Legal Status
Date of request for an examination (20060420)
Notification date of refusal decision (00000000)
Final disposal of an application (registration)
Date of final disposal of an application (20071029)
Patent registration number (1007952550000)
Date of registration (20080109)
Number of opposition against the grant of a patent ( )
Date of opposition against the grant of a patent (00000000)
Number of trial against decision to refuse ( )
Date of requesting trial against decision to refuse ( )

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년01월15일
(11) 등록번호 10-0795255
(24) 등록일자 2008년01월09일

(51) Int. Cl.
G11B 20/10 (2006.01)
(21) 출원번호 10-2001-7016423
(22) 출원일자 2001년12월21일
심사청구일자 2006년04월20일
번역문제출일자 2001년12월21일
(65) 공개번호 10-2002-0020919
(43) 공개일자 2002년03월16일
(86) 국제출원번호 PCT/JP2001/003414
국제출원일자 2001년04월20일
(87) 국제공개번호 WO 2001/82608
국제공개일자 2001년11월01일
(30) 우선권주장
JP-P-2000-00183770 2000년04월21일 일본(JP)
JP-P-2000-00268043 2000년09월05일 일본(JP)
(56) 선행기술조사문헌
JP2000-341646
JP11-273227
JP10-290432
전체 청구항 수 : 총 21 항

(73) 특허권자
소니 가부시끼 가이샤
일본국 도쿄도 미나토쿠 코난 1-7-1
(72) 발명자
가또,모또끼
일본141-0001도꾜도시나가와꾸기따시나가와6쪼메7-35소니가부시끼가이샤내
하마다,도시야
일본141-0001도꾜도시나가와꾸기따시나가와6쪼메7-35소니가부시끼가이샤내
(74) 대리인
구영창, 장수길

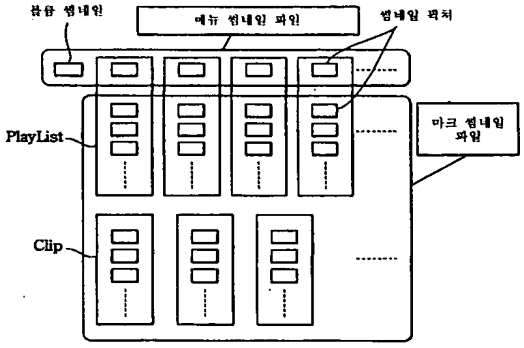
심사관 : 석상문

(54) 정보 처리 장치 및 방법, 프로그램과 기록 매체

(57) 요 약

입력된 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 AV 스트림을 관리하기 위한 관리 정보로서 생성함과 함께, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark를 생성하고, ClipMark 및 PlayListMark를 각각 독립한 테이블로서 기록 매체에 기록하도록 하였기 때문에, AV 스트림의 원하는 위치에, 신속하게 또한 확실하게 액세스하는 것이 가능해진다.

대표도 - 도13



(81) 지정국

국내특허 : 중국, 대한민국, 미국

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 독일, 덴마크, 스페인, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투칼, 스웨덴, 핀란드, 사이프러스, 터어키

특허청구의 범위

청구항 1

입력된 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 상기 AV 스트림을 관리하기 위한 관리 정보로서 생성함과 함께,

상기 AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 재생을 개시하는 위치를 가리키는 마크로 구성되는 PlayListMark를 생성하는 생성 수단과,

상기 ClipMark 및 PlayListMark를 각각 독립된 테이블로서 기록 매체에 기록하는 기록 수단

을 포함하는 정보 처리 장치.

청구항 2

제1항에 있어서,

상기 생성 수단은 상기 ClipMark를 ClipMarkInformation 파일로서 생성함과 함께, 상기 PlayList를 PlayList 파일로서 생성하는 정보 처리 장치.

청구항 3

제1항에 있어서,

상기 PlayListMark는 상기 PlayList를 재생할 때의 리쥼(resume)점을 나타내는 마크를 더 포함하는 정보 처리 장치.

청구항 4

제1항에 있어서,

상기 PlayList를 재생할 때, 상기 PlayList의 재생 구간에 대응하는 상기 AV 스트림의 ClipMark를 구성하는 상기 마크를 참조하는 정보 처리 장치.

청구항 5

제1항에 있어서,

상기 PlayListMark의 상기 마크는, 프레젠테이션 타임 스탬프와, 상기 PlayList의 재생 경로를 구성하는 상기 AV 스트림 데이터 상의 지정된 하나의 재생 구간을 나타내는 식별 정보를 포함하는 정보 처리 장치.

청구항 6

제1항에 있어서,

상기 ClipMark를 구성하는 상기 마크, 또는 상기 PlayListMark를 구성하는 상기 마크는 엘리먼트리 스트림의 엔트리 포인트를 특정하는 정보를 포함하는 정보 처리 장치.

청구항 7

제1항에 있어서,

상기 PlayListMark의 상기 마크는 사용자가 지정한 마음에 드는 장면의 개시점 또는 PlayList의 리쥼점을 적어도 포함하는 타입의 정보를 포함하는 정보 처리 장치.

청구항 8

제1항에 있어서,

상기 ClipMark를 구성하는 상기 마크와 상기 PlayListMark를 구성하는 상기 마크는, 상기 AV 스트림의 엔트리 포인트에 대응하는 상대적인 소스 패킷의 어드레스로 표시되는 정보 처리 장치.

청구항 9

제8항에 있어서,

상기 ClipMark를 구성하는 상기 마크와 상기 PlayListMark를 구성하는 상기 마크는, 상기 AV 스트림의 엔트리 포인트에 대응하는 상대적인 소스 패킷의 제1 어드레스와, 상기 제1 어드레스로부터의 오프셋의 어드레스인 제2 어드레스로 표시되는 정보 처리 장치.

청구항 10

제1항에 있어서,

상기 기록 수단에 의한 기록 시에 검출된 상기 특징적인 화상의 타입을 검출하는 타입 검출 수단을 더 포함하고,

상기 기록 수단은, 상기 ClipMark를 구성하는 상기 마크와, 상기 타입 검출 수단에 의해 검출된 상기 타입을 대응시켜 기록하는 정보 처리 장치.

청구항 11

제1항에 있어서,

상기 ClipMark의 상기 마크는, 장면 전환점, 커머셜의 개시점, 커머셜의 종료점, 또는 타이틀이 표시된 장면을 포함하는 정보 처리 장치.

청구항 12

정보 처리 방법에 있어서,

입력된 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를, 상기 AV 스트림을 관리하기 위한 관리 정보로서 생성함과 함께, 상기 AV 스트림 중의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 재생을 개시하는 위치를 가리키는 마크로 구성되는 PlayListMark를 생성하는 생성 단계와,

상기 ClipMark 및 PlayListMark를 각각 독립된 테이블로서 기록 매체에 기록할 때의 제어를 행하는 기록 제어 단계

를 포함하는 정보 처리 방법.

청구항 13

입력된 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를, 상기 AV 스트림을 관리하기 위한 관리 정보로서 생성함과 함께, 상기 AV 스트림 중의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 재생을 개시하는 위치를 가리키는 마크로 구성되는 PlayListMark를 생성하는 생성 단계와,

상기 ClipMark 및 PlayListMark를 각각 독립된 테이블로서 기록 매체에 기록할 때의 제어를 행하는 기록 제어 단계

를 포함하는 컴퓨터가 판독 가능한 프로그램이 기록되어 있는 기록 매체.

청구항 14

삭제

청구항 15

AV 스트림, AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark 및 상기 AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 재생을 개시하는 위치를 가리키는 마크로 구성되는 PlayListMark가 기록된 기록 매체를 재생하는 정보 처리 장치로서,

상기 기록 매체를 재생하는 재생 수단과,

재생된 상기 ClipMark 또는 PlayListMark에 기술된 마크에 대응하는 기록 위치를 취득함과 함께, 그 취득된 기록 위치에 따라 상기 재생 수단을 제어하는 제어 수단

을 포함하는 정보 처리 장치.

청구항 16

제15항에 있어서,

상기 PlayListMark에 대응하는 썸네일 화상에 의한 리스트를 사용자에게 제시하도록 제어하는 제시 제어 수단을 더 포함하는 정보 처리 장치.

청구항 17

제15항에 있어서,

상기 ClipMark를 구성하는 상기 마크와 상기 PlayListMark를 구성하는 상기 마크는, 상기 AV 스트림의 엔트리 포인트에 대응하는 상대적인 소스 패킷의 어드레스로 나타내는 정보 처리 장치.

청구항 18

제17항에 있어서,

상기 ClipMark를 구성하는 상기 마크와 상기 PlayListMark를 구성하는 상기 마크는, 상기 AV 스트림의 엔트리 포인트에 대응하는 상대적인 소스 패킷의 제1 어드레스와, 상기 제1 어드레스로부터의 오프셋의 어드레스인 제2 어드레스로 나타내는 정보 처리 장치.

청구항 19

제15항에 있어서,

상기 ClipMark의 상기 마크는 장면 전환점, 커머셜의 개시점, 커머셜의 종료점, 또는 타이틀이 표시된 장면을 포함하는 정보 처리 장치.

청구항 20

AV 스트림, AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark 및 상기 AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 재생을 개시하는 위치를 가리키는 마크로 구성되는 PlayListMark가 기록된 기록 매체를 재생하는 정보 처리 방법으로서,

상기 기록 매체를 재생하는 재생 단계와,

재생된 상기 ClipMark 또는 PlayListMark에 기술된 마크에 대응하는 기록 위치를 취득함과 함께, 그 취득된 기록 위치에 따라 재생 위치를 제어하는 제어 단계

를 포함하는 정보 처리 방법.

청구항 21

AV 스트림, AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark 및 상기 AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 재생을 개시하는 위치를 가리키는 마크로 구성되는 PlayListMark가 기록된 기록 매체를 재생하기 위한 컴퓨터가 판독 가능한 프로그램이 기록되어 있는 기록 매체로서,

상기 기록 매체를 재생하는 재생 단계와,

재생된 상기 ClipMark 또는 PlayListMark에 기술된 마크에 대응하는 기록 위치를 취득함과 함께, 그 취득된 기록 위치에 따라 재생 위치를 제어하는 제어 단계

를 포함하는 컴퓨터가 판독 가능한 프로그램이 기록되어 있는 기록 매체.

청구항 22

삭제

청구항 23

AV 스트림이 기록됨과 함께, 그 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark 및 상기 AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 재생을 개시하는 위치를 가리키는 마크로 구성되는 PlayListMark가 각각 독립된 테이블로서 기록되어 있는 기록 매체.

명 세 서

기 술 분 야

<1> 본 발명은 정보 처리 장치 및 방법, 프로그램과 기록 매체에 관한 것으로, 특히, AV 스트림의 원하는 위치에 신속하게 액세스할 수 있도록 한 정보 처리 장치 및 방법, 프로그램과 기록 매체에 관한 것이다.

배 경 기 술

<2> 최근, 기록 가능하며 재생 장치로부터 착탈 가능한 디스크형의 기록 매체로서, 각종 광 디스크가 제안되고 있다. 이러한 기록 가능한 광 디스크는 수 GB의 대용량 미디어로서 제안되고 있어, 비디오 신호 등의 AV(Audio Visual) 신호를 기록하는 미디어로서의 기대가 높다.

<3> 이 기록 가능한 광 디스크에 기록하는 디지털의 AV 신호의 소스(공급원)로서는 기록 장치 자신이 아날로그 입력의 오디오 비디오 신호를 MPEG-2 방식으로 화상 압축하여 만드는 비트 스트림이나 디지털 텔레비전 방송의 전파로부터 직접 얻을 수 있는 MPEG-2 방식의 비트 스트림 등이 있다.

<4> 일반적으로, 디지털 텔레비전 방송에서는 MPEG-2 전송 스트림이 사용된다. 전송 스트림은 전송 패킷이 연속된 스트림이며, 전송 패킷은 예를 들면, MPEG-2 비디오 스트림이나 MPEG-1 오디오 스트임이 패킷화된 것이다. 하나의 전송 패킷의 데이터 길이는 188바이트이다. 디지털 텔레비젼 방송에서 송신되는 전송 스트림의 AV 프로그램을 기록 장치에서 광 디스크에 그대로 기록하면 비디오나 오디오의 품질을 전혀 열화시키지 않고 기록하는 것이 가능하다.

<5> 사용자가 광 디스크에 기록되어 있는 전송 스트림 중에서 흥미가 있는 장면,예를 들면 프로그램의 검색점 등을 서치할 수 있도록 하기 위해, 재생 장치는 랜덤 액세스 재생하도록 요구된다.

<6> 일반적으로, MPEG-2 비디오 스트림은 0.5초 정도의 간격으로 I픽쳐를 부호화하고, 그 이하의 픽쳐는 P픽쳐 또는 B픽쳐로서 부호화된다. 따라서, MPEG-2 비디오의 스트림이 기록된 광 디스크로부터 랜덤 액세스하고, 비디오 재생하는 경우, 우선 I픽쳐를 서치해야 한다.

<7> 그러나, 종래는 광 디스크에 기록되어 있는 전송 스트림에 랜덤 액세스하고, 비디오 재생하는 경우에, I픽쳐의 개시 바이트를 효율적으로 서치하는 것이 곤란하였다. 즉, 광 디스크 상의 전송 스트림의 랜덤 바이트 위치로부터, 판독된 비디오 스트림의 신택스를 해석하고, I픽쳐의 개시 바이트를 서치해야 하므로, I픽쳐의 서치에 시간이 소요되어 사용자로부터의 입력에 대하여 응답이 빠른 랜덤 액세스 재생을 행하는 것이 곤란하였다.

<8> <발명의 개시>

<9> 본 발명의 목적은 이러한 상황을 감안하여 이루어진 것으로, 사용자에 대한 랜덤 액세스 재생의 지시에 대하여, 기록 매체로부터의 전송 스트림의 판독 위치의 결정과 스트림의 복호 개시를 빠르게 행할 수 있도록 하는 것에 있다.

<10> 본 발명에 따른 정보 처리 장치는, 입력된 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 AV 스트림을 관리하기 위한 관리 정보로서 생성함과 함께, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark를 생성하는 생성 수단과, ClipMark 및 PlayListMark를 각각 독립된 테이블로서 기록 매체에 기록하는 기록 수단을 갖는다.

<11> 여기서, 생성 수단은 ClipMark를 ClipMarkInformation 파일로서 생성함과 함께, PlayList를 PlayList 파일로서 생성하도록 할 수 있다.

<12> PlayListMark는 PlayList를 재생할 때의 리쥼점을 나타내는 마크를 더 포함하도록 할 수 있다.

<13> PlayList를 재생할 때, PlayList의 재생 구간에 대응하는 AV 스트림의 ClipMark를 구성하는 마크를 참조하도록 할 수 있다.

<14> PlayListMark의 마크는 프레젠테이션 타임 스탬프와, PlayList의 재생 경로를 구성하는 AV 스트림 데이터 상의 지정된 하나의 재생 구간을 나타내는 식별 정보를 포함하도록 할 수 있다.

<15> ClipMark를 구성하는 마크와 PlayListMark를 구성하는 마크는 엘리먼트리 스트림의 엔트리 포인트를 특정하는 정보를 포함하도록 할 수 있다.

<16> PlayListMark의 마크는 사용자가 지정한 마음에 드는 장면의 개시점 또는 PlayList의 리쥼점을 적어도 포함하는 타입의 정보를 포함하도록 할 수 있다.

<17> ClipMark를 구성하는 마크와 PlayListMark를 구성하는 마크는 AV 스트림의 엔트리 포인트에 대응하는 상대적인 소스 패킷의 어드레스로 나타내도록 할 수 있다.

<18> ClipMark를 구성하는 마크와 PlayListMark를 구성하는 마크는 AV 스트림의 엔트리 포인트에 대응하는 상대적인 소스 패킷의 제1 어드레스와, 제1 어드레스로부터의 오프셋의 어드레스인 제2 어드레스로 나타내도록 할 수 있다.

<19> 기록 수단에 의한 기록을 행할 때에 검출된 특징적인 화상의 타입을 검출하는 타입 검출 수단을 더 포함하고, 기록 수단은 ClipMark를 구성하는 마크와, 타입 검출 수단에 의해 검출된 타입을 대응시켜 기록하도록 할 수 있다.

<20> ClipMark의 마크는 장면 전환점, 커머셜의 개시점, 커머셜의 종료점, 또는 타이틀이 표시된 장면을 포함하도록 할 수 있다.

<21> 본 발명에 따른 정보 처리 방법은, 입력된 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 AV 스트림을 관리하기 위한 관리 정보로서 생성함과 함께, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark를 생성하는 생성 단계와, ClipMark 및 PlayListMark를 각각 독립된 테이블로서 기록 매체에 기록하는 할 때의 제어를 행하는 기록 제어 단계를 포함한다.

<22> 본 발명에 따른 기록 매체의 프로그램은, 입력된 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 AV 스트림을 관리하기 위한 관리 정보로서 생성함과 함께, AV 스트림 내의 소정의 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark를 생성하는 생성 단계와, ClipMark 및 PlayListMark를 각각 독립된 테이블로서 기록 매체에 기록할 때의 제어를 행하는 기록 제어 단계를 포함한다.

<23> 본 발명에 따른 프로그램은, 입력된 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 AV 스트림을 관리하기 위한 관리 정보로서 생성함과 함께, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark를 생성하는 생성 단계와, ClipMark 및 PlayListMark를 각각 독립된 테이블로서 기록 매체에 기록할 때의 제어를 행하는 기록 제어 단계를 컴퓨터에 실행시킨다.

<24> 본 발명의 정보 처리 장치는, AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 포함하는 AV 스트림을 관리하기 위한 관리 정보와, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark를 판독하는 판독 수단과, 판독 수단에 의해 판독된 관리 정보와 PlayListMark에 의한 정보를 제시하는 제시 수단과, 제시 수단에 의해 제시된 정보로부터 사용자가 재생을 지시한 PlayList에 대응하는 ClipMark를 참조하는 참조 수단과, 참조 수단에 의해 참조된 ClipMark를 포함하며, ClipMark에 대응하는 위치로부터 AV 스트림을 재생하는 재생 수단을 포함한다.

<25> 여기서, 제시 수단은 PlayListMark에 대응하는 썸네일 화상에 의한 리스트를 사용자에게 제시하도록 할 수 있다.

<26> ClipMark를 구성하는 마크와 PlayListMark를 구성하는 마크는 AV 스트림의 엔트리 포인트에 대응하는 상대적인 소스 패킷의 어드레스로 나타내도록 할 수 있다.

<27> ClipMark를 구성하는 마크와 PlayListMark를 구성하는 마크는 AV 스트림의 엔트리 포인트에 대응하는 상대적인

소스 패킷의 제1 어드레스와, 제1 어드레스로부터의 오프셋의 어드레스인 제2 어드레스로 나타내도록 할 수 있다.

<28> ClipMark의 마크는 장면 전환점, 커머셜의 개시점, 커머셜의 종료점, 또는 타이틀이 표시된 장면을 포함하도록 할 수 있다.

<29> 본 발명에 따른 정보 처리 방법은, AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 포함하는 AV 스트림을 관리하기 위한 관리 정보와, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark의 판독을 제어하는 판독 제어 단계와, 판독 제어 단계의 처리에서 판독이 제어된 관리 정보와 PlayListMark에 의한 정보를 제시하는 제시 단계와, 제시 단계의 처리에서 제시된 정보로부터 사용자가 재생을 지시한 PlayList에 대응하는 ClipMark를 참조하는 참조 단계와, 참조 단계의 처리에서 참조된 ClipMark를 포함하며, ClipMark에 대응하는 위치로부터의 AV 스트림의 재생을 제어하는 재생 제어 단계를 포함한다.

<30> 본 발명에 따른 기록 매체의 프로그램은, AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 포함하는 AV 스트림을 관리하기 위한 관리 정보와, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark의 판독을 제어하는 판독 제어 단계와, 판독 제어 단계의 처리에서 판독이 제어된 관리 정보와 PlayListMark에 의한 정보를 제시하는 제시 단계와, 제시 단계의 처리에서 제시된 정보로부터 사용자가 재생을 지시한 PlayList에 대응하는 ClipMark를 참조하는 참조 단계와, 참조 단계의 처리에서 참조된 ClipMark를 포함하며, ClipMark에 대응하는 위치로부터의 AV 스트림의 재생을 제어하는 재생 제어 단계를 포함한다.

<31> 본 발명에 따른 프로그램은, AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 포함하는 AV 스트림을 관리하기 위한 관리 정보와, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark의 판독을 제어하는 판독 제어 단계와, 판독 제어 단계의 처리에서 판독이 제어된 관리 정보와 PlayListMark에 의한 정보를 제시하는 제시 단계와, 제시 단계의 처리에서 제시된 정보로부터, 사용자가 재생을 지시한 PlayList에 대응하는 ClipMark를 참조하는 참조 단계와, 참조 단계의 처리에서 참조된 ClipMark를 포함하며, ClipMark에 대응하는 위치로부터의 AV 스트림의 재생을 제어하는 재생 제어 단계를 컴퓨터에 실행시킨다.

<32> 본 발명에 따른 기록 매체에는, AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 포함하는 AV 스트림을 관리하기 위한 관리 정보와, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark가 각각 독립된 테이블로서 기록되어 있다.

<33> 본 발명에 따른 정보 처리 장치 및 방법, 및 프로그램에 있어서는, 입력된 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 AV 스트림을 관리하기 위한 관리 정보로서 생성함과 함께, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark가 생성되고, ClipMark 및 PlayListMark가 각각 독립된 테이블로서 기록 매체에 기록된다.

<34> 본 발명에 따른 정보 처리 장치 및 방법, 및 프로그램은 AV 스트림으로부터 추출된 특징적인 화상을 가리키는 마크로 구성되는 ClipMark를 포함하는 AV 스트림을 관리하기 위한 관리 정보와, AV 스트림 내의 소정 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로 지정한 화상을 가리키는 마크로 구성되는 PlayListMark가 판독되고, 그 판독된 관리 정보와 PlayListMark에 의한 정보가 제시되며, 제시된 정보로부터 사용자가 재생을 지시한 PlayList에 대응하는 ClipMark가 참조되고, 참조된 ClipMark를 포함하며 ClipMark에 대응하는 위치로부터 AV 스트림이 재생된다.

<35> 본 발명의 또 다른 목적, 특징이나 이점은 후술하는 본 발명의 실시예나 첨부하는 도면에 기초하는 보다 상세한 설명으로부터 분명해질 것이다.

산업상 이용 가능성

<673> 이상과 같이 본 발명에 따른 정보 처리 장치 및 방법, 및 프로그램에서는, 입력된 AV 스트림으로부터 추출된 특징적인 화상을 지시하는 마크로 구성되는 ClipMark를, AV 스트림을 관리하기 위한 관리 정보로서 생성함과 함께, AV 스트림 중의 소정의 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간 중에서, 사용자가 임의로

지정한 화상을 지시하는 마크로 구성되는 PlayListMark를 생성하고, ClipMark 및 PlayListMark를 각각 독립한 테이블로서 기록 매체에 기록하도록 하였기 때문에, AV 스트림의 원하는 위치에, 신속 또한 확실하게 액세스하는 것이 가능해진다.

<674> 또한, 본 발명에 따른 정보 처리 장치 및 방법, 및 프로그램은, AV 스트림으로부터 추출된 특징적인 화상을 지시하는 마크로 구성되는 ClipMark를 포함하는 AV 스트림을 관리하기 위한 관리 정보와, AV 스트림 중의 소정의 구간의 조합을 정의하는 PlayList에 대응하는 재생 구간의 중으로부터, 사용자가 임의로 지정한 화상을 지시하는 마크로 구성되는 PlayListMark를 판독하고, 그 판독된 관리 정보와 PlayListMark에 의한 정보를 제시하여, 제시된 정보로부터, 사용자가 재생을 지시한 PlayList에 대응하는 ClipMark을 참조하여, 참조된 ClipMark를 포함하고, ClipMark에 대응하는 위치로부터 AV 스트림을 재생하도록 하였기 때문에, AV 스트림의 원하는 위치에, 신속하고 확실하게 액세스하는 것이 가능해진다.

도면의 간단한 설명

<36> 도 1은 본 발명을 적용한 기록 재생 장치의 구성을 나타내는 도면.

<37> 도 2는 기록 재생 장치(1)에 의해 기록 매체에 기록되는 데이터의 포맷에 대하여 설명하는 도면.

<38> 도 3은 Real PlayList와 Virtual PlayList에 대하여 설명하는 도면.

<39> 도 4의 A~C는 Real PlayList의 작성에 대하여 설명하는 도면.

<40> 도 5의 A~C는 Real PlayList의 삭제에 대하여 설명하는 도면.

<41> 도 6의 A 및 B는 어셈블 편집에 대하여 설명하는 도면.

<42> 도 7은 Virtual PlayList에 서브 패스를 설치하는 경우에 대해 설명하는 도면.

<43> 도 8은 PlayList의 재생 순서의 변경에 대하여 설명하는 도면.

<44> 도 9는 PlayList 상의 마크와 Clip 상의 마크에 대하여 설명하는 도면.

<45> 도 10은 메뉴 썸네일에 대하여 설명하는 도면.

<46> 도 11은 PlayList에 부가되는 마크에 대하여 설명하는 도면.

<47> 도 12는 클립에 부가되는 마크에 대하여 설명하는 도면.

<48> 도 13은 PlayList, Clip, 썸네일 파일의 관계에 대하여 설명하는 도면.

<49> 도 14는 디렉토리 구조에 대하여 설명하는 도면.

<50> 도 15는 info.dvr의 신택스를 나타내는 도면.

<51> 도 16은 DVR Volume의 신택스를 나타내는 도면.

<52> 도 17은 ResumeVolume의 신택스를 나타내는 도면.

<53> 도 18은 UIAppInfoVolume의 신택스를 나타내는 도면.

<54> 도 19는 Charactersetvalue의 테이블을 나타내는 도면.

<55> 도 20은 TabldOfPlayList의 신택스를 나타내는 도면.

<56> 도 21은 TableOfPlayList의 다른 신택스를 나타내는 도면.

<57> 도 22는 MakersPrivateData의 신택스를 나타내는 도면.

<58> 도 23은 xxxxx.rpls와 yyyyy.vpls의 신택스를 나타내는 도면.

<59> 도 24의 A~C는 PlayList에 대하여 설명하는 도면.

<60> 도 25는 PlayList의 신택스를 나타내는 도면.

<61> 도 26은 PlayList_type의 테이블을 나타내는 도면.

<62> 도 27은 UIAppinfoPlayList의 신택스를 나타내는 도면.

<63> 도 28의 A~C는 도 27에 도시한 UIAppinfoPlayList의 신택스 내의 플래그에 대하여 설명하는 도면.

<64> 도 29는 PlayItem에 대하여 설명하는 도면.

<65> 도 30은 PlayItem에 대하여 설명하는 도면.

<66> 도 31은 PlayItem에 대하여 설명하는 도면.

<67> 도 32는 PlayItem의 신택스를 나타내는 도면.

<68> 도 33은 IN_time에 대하여 설명하는 도면.

<69> 도 34는 OUT_time에 대하여 설명하는 도면.

<70> 도 35는 Connection_Condition의 테이블을 나타내는 도면.

<71> 도 36의 A~D는 Connection_Condition에 대하여 설명하는 도면.

<72> 도 37은 BridgeSequenceInfo를 설명하는 도면.

<73> 도 38은 BridgeSequenceInfo의 신택스를 나타내는 도면.

<74> 도 39는 SubPlayItem에 대하여 설명하는 도면.

<75> 도 40은 SubPlayItem의 신택스를 나타내는 도면.

<76> 도 41은 SubPath_type의 테이블을 나타내는 도면.

<77> 도 42는 PlayListMark의 신택스를 나타내는 도면.

<78> 도 43은 Mark_type의 테이블을 나타내는 도면.

<79> 도 44는 Mark_time_stamp를 설명하는 도면.

<80> 도 45는 zzzzz.clip의 신택스를 나타내는 도면.

<81> 도 46은 ClipInfo의 신택스를 나타내는 도면.

<82> 도 47은 Clip_stream_type의 테이블을 나타내는 도면.

<83> 도 48은 offset_SPN에 대하여 설명하는 도면.

<84> 도 49는 offset_SPN에 대하여 설명하는 도면.

<85> 도 50의 A 및 B는 STC 구간에 대하여 설명하는 도면.

<86> 도 51은 STC_Info에 대하여 설명하는 도면.

<87> 도 52는 STC_info의 신택스를 나타내는 도면.

<88> 도 53은 ProgramInfo를 설명하는 도면.

<89> 도 54는 ProgramInfo의 신택스를 나타내는 도면.

<90> 도 55는 VideoCodingInfo의 신택스를 나타내는 도면.

<91> 도 56은 Video_format의 테이블을 나타내는 도면.

<92> 도 57은 frame_rate의 테이블을 나타내는 도면.

<93> 도 58은 display_aspect_ratio의 테이블을 나타내는 도면.

<94> 도 59는 AudioCodingInfo의 신택스를 나타내는 도면.

<95> 도 60은 audio_coding의 테이블을 나타내는 도면.

<96> 도 61은 audio_component_type의 테이블을 나타내는 도면.

<97> 도 62는 sampling_frequency의 테이블을 나타내는 도면.

<98> 도 63은 CPI에 대하여 설명하는 도면.

<99> 도 64는 CPI에 대하여 설명하는 도면.

<100> 도 65는 CPI의 신택스를 나타내는 도면.

<101> 도 66은 CPI_type의 테이블을 나타내는 도면.

<102> 도 67은 비디오 EP_map에 대하여 설명하는 도면.

<103> 도 68은 EP_map에 대하여 설명하는 도면.

<104> 도 69는 EP_map에 대하여 설명하는 도면.

<105> 도 70은 EP_map의 신택스를 나타내는 도면.

<106> 도 71은 EP_type values의 테이블을 나타내는 도면.

<107> 도 72는 EP_map_for_one_stream_PID의 신택스를 나타내는 도면.

<108> 도 73은 TU_map에 대하여 설명하는 도면.

<109> 도 74는 TU_map의 신택스를 나타내는 도면.

<110> 도 75는 ClipMark의 신택스를 나타내는 도면.

<111> 도 76은 mark_type의 테이블을 나타내는 도면.

<112> 도 77은 mark_type_stamp의 테이블을 나타내는 도면.

<113> 도 78은 ClipMark의 신택스의 다른 예를 나타내는 도면.

<114> 도 79는 mark_type의 테이블의 다른 예를 나타내는 도면.

<115> 도 80은 mark_entry()와 representative_picture_entry()의 예를 나타내는 도면.

<116> 도 81은 mark_entry()와 representative_picture_entry()의 신택스를 나타내는 도면.

<117> 도 82는 mark_entry()와 representative_picture_entry()의 신택스의 다른 예를 나타내는 도면.

<118> 도 83은 RSPN_ref_EP_start와 offset_num_pictures의 관계를 설명하는 도면.

<119> 도 84는 mark_entry()와 representative_picture_entry()의 신택스의 다른 예를 나타내는 도면.

<120> 도 85는 ClipMark와 EP_map의 관계를 설명하는 도면.

<121> 도 86은 menu.thmb와 mark.thmb의 신택스를 나타내는 도면.

<122> 도 87은 Thumbnail의 신택스를 나타내는 도면.

<123> 도 88은 thumbnail_picture_format의 테이블을 나타내는 도면.

<124> 도 89A 및 도 89B는 tn_block에 대하여 설명하는 도면.

<125> 도 90은 DVR MPEG2의 전송 스트림의 구조에 대하여 설명하는 도면.

<126> 도 91은 DVR MPEG2의 전송 스트림의 레코더 모델을 나타내는 도면.

<127> 도 92는 DVR MPEG2의 전송 스트림의 플레이어 모델을 나타내는 도면.

<128> 도 93은 소스 패킷(source packet)의 신택스를 나타내는 도면.

<129> 도 94는 TP_extra_header의 신택스를 나타내는 도면.

<130> 도 95는 copy permission indicator의 테이블을 나타내는 도면.

<131> 도 96은 심리스(seamless) 접속에 대하여 설명하는 도면.

<132> 도 97은 심리스 접속에 대하여 설명하는 도면.

<133> 도 98은 심리스 접속에 대하여 설명하는 도면.

<134> 도 99는 심리스 접속에 대하여 설명하는 도면.

<135> 도 100은 심리스 접속에 대하여 설명하는 도면.

<136> 도 101은 오디오의 오버랩에 대하여 설명하는 도면.

<137> 도 102는 BridgeSequence를 이용한 심리스 접속에 대하여 설명하는 도면.

<138> 도 103은 BridgeSequence를 이용하지 않은 심리스 접속에 대하여 설명하는 도면.

<139> 도 104는 DVR STD 모델을 나타내는 도면.

<140> 도 105는 복호, 표시의 타이밍차트를 나타내는 도면.

<141> 도 106은 도 81의 신택스의 경우에서의 마크점으로 나타내는 장면의 검색 재생을 설명하는 흐름도.

<142> 도 107은 도 81의 신택스의 경우에서의 재생 동작을 설명하는 도면.

<143> 도 108은 EP_map의 예를 나타내는 도면.

<144> 도 109는 ClipMark의 예를 나타내는 도면.

<145> 도 110은 도 81의 신택스의 경우에서의 CM 스킵 재생 처리를 설명하는 흐름도.

<146> 도 111은 도 81의 신택스의 경우에서의 CM 스킵 재생 처리를 설명하는 흐름도.

<147> 도 112는 도 82의 신택스의 경우에서의 마크점으로 나타내는 장면의 프로그램 개시 위치 검출 재생을 설명하는 흐름도.

<148> 도 113은 도 82의 신택스의 경우에서의 재생을 설명하는 도면.

<149> 도 114는 EP_map의 예를 나타내는 도면.

<150> 도 115는 ClipMark의 예를 나타내는 도면.

<151> 도 116은 도 82의 신택스의 경우에서의 CM 스킵 재생을 설명하는 흐름도.

<152> 도 117은 도 82의 신택스의 경우에서의 CM 스킵 재생을 설명하는 흐름도.

<153> 도 118은 도 84의 신택스의 경우에서의 마크점으로 나타내는 장면의 검색 재생을 설명하는 흐름도.

<154> 도 119는 도 84의 신택스의 경우에서의 재생을 설명하는 도면.

<155> 도 120은 EP_map의 예를 나타내는 도면.

<156> 도 121은 CIjpMark의 예를 나타내는 도면.

<157> 도 122는 도 84의 신택스의 경우에서의 CM 스킵 재생을 설명하는 흐름도.

<158> 도 123은 도 84의 신택스의 경우에서의 CM 스킵 재생을 설명하는 흐름도.

<159> 도 124는 어플리케이션 포맷을 나타내는 도면.

<160> 도 125는 PlayList 상의 마크와 Clip 상의 마크를 설명하는 도면.

<161> 도 126은 ClipMark의 신택스의 다른 예를 나타내는 도면.

<162> 도 127은 ClipMark의 신택스의 또 다른 예를 나타내는 도면.

<163> 도 128은 아날로그 AV 신호를 인코드하여 기록하는 경우의 ClipMark의 작성에 대하여 설명하는 흐름도.

<164> 도 129는 전송 스트림을 기록하는 경우의 ClipMark의 작성에 대하여 설명하는 흐름도.

<165> 도 130은 RealPlayList의 작성에 대하여 설명하는 흐름도.

<166> 도 131은 VirtualPlayList의 작성에 대하여 설명하는 흐름도.

<167> 도 132는 PlayList의 재생에 대하여 설명하는 흐름도.

<168> 도 133은 PlayListMark의 작성에 대하여 설명하는 흐름도.

<169> 도 134는 PlayList를 재생할 때의 검색 재생에 대하여 설명하는 흐름도.

<170> 도 135는 PlayListMark의 신택스를 나타내는 도면.

<171> 도 136은 PlayListMark의 Mark_Type을 설명하기 위한 도면.

<172> 도 137은 ClipMark의 다른 신택스를 나타내는 도면.

<173> 도 138은 ClipMark의 Mark_type을 설명하기 위한 도면.

<174> 도 139는 매체를 설명하는 도면.

<175> <발명을 실시하기 위한 최량의 형태>

<176> 이하에, 본 발명의 실시예에 대하여 도면을 참조하여 설명한다. 도 1은 본 발명을 적용한 기록 재생 장치(1)의 내부 구성예를 나타내는 도면이다. 우선, 외부로부터 입력된 신호를 기록 매체에 기록하는 동작을 행하는 부분의 구성에 대하여 설명한다. 기록 재생 장치(1)는 아날로그 데이터, 또는, 디지털 데이터를 입력하고, 기록할 수 있는 구성으로 되어 있다.

<177> 단자(11)에는 아날로그의 비디오 신호가, 단자(12)에는 아날로그의 오디오 신호가 각각 입력된다. 단자(11)에 입력된 비디오 신호는 해석부(14)와 AV 인코더(15)로 각각 출력된다. 단자(12)에 입력된 오디오 신호는 AV 인코더(15)로 출력된다. 해석부(14)는 입력된 비디오 신호로부터 장면 전환 등의 특징점을 추출한다.

<178> AV 인코더(15)는 입력된 비디오 신호와 오디오 신호를 각각 부호화하고, 부호화 비디오 스트림(V), 부호화 오디오 스트림(A) 및 AV 동기 등의 시스템 정보(S)를 멀티플렉서(16)로 출력한다.

<179> 부호화 비디오 스트림은, 예를 들면, MPEG(Moving Picture Expert Group)2 방식에 의해 부호화된 비디오 스트림으로, 부호화 오디오 스트림은, 예를 들면, MPEG1 방식에 의해 부호화된 오디오 스트림이나, 돌비 AC3 방식(상표)에 의해 부호화된 오디오 스트림 등이다. 멀티플렉서(16)는 입력된 비디오 및 오디오의 스트림을 입력 시스템 정보에 기초하여 다중화하여, 스위치(17)를 통해 다중화 스트림 해석부(18)와 소스 패킷타이저(source packetizer; 19)로 출력한다.

<180> 다중화 스트림은, 예를 들면, MPEG2 전송 스트림이나 MPEG2 프로그램 스트림이다. 소스 패킷타이저(19)는 입력된 다중화 스트림을, 이 스트림을 기록시키는 기록 매체(100)의 어플리케이션 포맷에 따라 소스 패킷으로 구성되는 AV 스트림을 부호화한다. AV 스트림은 ECC(오류 정정) 부호화부(20), 변조부(21)에서 소정의 처리가 실시되어, 기입부(22)로 출력된다. 기입부(22)는 제어부(23)로부터 출력되는 제어 신호에 기초하여, 기록 매체(100)에 AV 스트림 파일을 기입한다(기록한다).

<181> 디지털 인터페이스 또는 디지털 텔레비전 튜너로부터 입력되는 디지털 텔레비전 방송 등의 전송 스트림은 단자(13)에 입력된다. 단자(13)에 입력된 전송 스트림의 기록 방식에는 2가지가 있으며, 이들은 트랜스페어런트하게 기록하는 방식과, 기록 비트 레이트를 저하시키는 등의 목적을 위해 재차 인코드를 행한 후에 기록하는 방식이다. 기록 방식의 지시 정보는 사용자 인터페이스로서의 단자(24)로부터 제어부(23)에 입력된다.

<182> 입력 전송 스트림을 트랜스페어런트하게 기록하는 경우, 단자(13)에 입력된 전송 스트림은 스위치(17)를 통해 다중화 스트림 해석부(18)와, 소스 패킷타이저(19)로 출력된다. 이 이후의 기록 매체(100)로 AV 스트림이 기록되기까지의 처리는 상술한 입력 오디오 신호와 비디오 신호를 부호화하여 기록하는 경우와 동일한 처리이기 때문에, 그에 대한 설명은 생략한다.

<183> 입력 전송 스트림을 재차 인코드한 후에 기록하는 경우, 단자(13)에 입력된 전송 스트림은 디멀티플렉서(26)에 입력된다. 디멀티플렉서(26)는 입력된 전송 스트림에 대하여 디멀티플렉스 처리를 실시하고, 비디오 스트림(V), 오디오 스트림(A) 및 시스템 정보(S)를 추출한다.

<184> 디멀티플렉서(26)에 의해 추출된 스트림(정보) 중, 비디오 스트림은 AV 디코더(27)로, 오디오 스트림과 시스템 정보는 멀티플렉서(16)로 각각 출력된다. AV 디코더(27)는 입력된 비디오 스트림을 복호하고, 그 재생 비디오 신호를 AV 인코더(15)로 출력한다. AV 인코더(15)는 입력 비디오 신호를 부호화하고, 부호화 비디오 스트림(V)을 멀티플렉서(16)로 출력한다.

<185> 한편, 디멀티플렉서(26)로부터 출력되어, 멀티플렉서(16)에 입력된 오디오 스트림과 시스템 정보 및 AV 인코더(15)로부터 출력된 비디오 스트림은 입력 시스템 정보에 기초하여 다중화되어, 다중화 스트림으로서 다중화 스트림 해석부(18)와 소스 패킷타이저(19)로 스위치(17)를 통해 출력된다. 이 이후의 기록 매체(100)에 AV 스트림이 기록되기까지의 처리는 상술한 입력 오디오 신호와 비디오 신호를 부호화하여 기록하는 경우와 동일한 처

리이기 때문에, 그에 대한 설명은 생략한다.

<186> 이상과 같은 기록 재생 장치(1)는 AV 스트림의 파일을 기록 매체(100)에 기록함과 함께, 그 파일을 설명하는 어플리케이션 데이터 베이스 정보도 기록한다. 어플리케이션 데이터 베이스 정보는 제어부(23)에 의해 작성된다. 제어부(23)로의 입력 정보는 해석부(14)로부터의 동화상의 특징 정보, 다중화 스트림 해석부(18)로부터의 AV 스트림의 특징 정보 및 단자(24)로부터 입력되는 사용자로부터의 지시 정보이다.

<187> 해석부(14)로부터 공급되는 동화상의 특징 정보는 AV 인코더(15)가 비디오 신호를 부호화하는 경우에서, 해석부(14)에 의해 생성된 것이다. 해석부(14)는 입력 비디오 신호와 오디오 신호의 내용을 해석하여, 입력 동화상 신호 중의 특징적인 화상(클립마크)에 관계하는 정보를 생성한다. 이것은, 예를 들면 입력 비디오 신호 중의 프로그램의 개시점, 장면 전환점, CM 커머셜의 개시점, 종료점, 타이틀이나 텔롭 등의 특징적 클립마크점의 화상의 지시 정보이며, 또한, 거기에는 그 화상의 썸네일도 포함된다. 또한 오디오 신호의 스테레오와 모노럴의 전환점이나, 무음 구간 등의 정보도 포함된다.

<188> 이들 화상의 지시 정보는 제어부(23)를 통해 멀티플렉서(16)로 입력된다. 멀티 플렉서(16)는 제어부(23)로부터 클립마크로서 지정되는 부호화 픽쳐를 다중화할 때, 그 부호화 픽쳐를 AV 스트림 상에서 특정하기 위한 정보를 제어부(23)로 되돌린다. 구체적으로는, 이 정보는 픽쳐의 PTS(프레젠테이션 타임 스탬프) 또는 그 부호화 픽쳐의 AV 스트림 상에서의 어드레스 정보이다. 제어부(24)는 특징적인 화상의 종류와 그 부호화 픽쳐를 AV 스트림 사아에서 특정하기 위한 정보를 관련지어 기억한다.

<189> 다중화 스트림 해석부(18)로부터의 AV 스트림의 특징 정보는 기록되는 AV 스트림의 부호화 정보에 관계되는 정보로, 해석부(18)에 의해 생성된다. 예를 들면, AV 스트림 내의 I픽쳐의 타임 스탬프와 어드레스 정보, 시스템 타임 클럭의 불연속점 정보, AV 스트림의 부호화 파라미터, AV 스트림 중의 부호화 파라미터의 변화점 정보 등이 포함된다. 또한, 단자(13)로부터 입력되는 전송 스트림을 트랜스페어런트하게 기록하는 경우, 다중화 스트림 해석부(18)는 입력 전송 스트림 중으로부터 앞에 나온 클립마크를 검출하고, 그 종류와 클립마크로 지정하는 픽쳐를 특정하기 위한 정보를 생성한다.

<190> 단자(24)로부터의 사용자의 지시 정보는 AV 스트림 중, 사용자가 지정한 재생 구간의 지정 정보, 그 재생 구간의 내용을 설명하는 캐릭터 문자, 사용자가 좋아하는 장면으로 세트하는 북 마크나 리즘점의 정보 등이다.

<191> 제어부(23)는 상기한 입력 정보에 기초하여, AV 스트림의 데이터 베이스(Clip), AV 스트림의 재생 구간(PlayItem)을 그룹화한 것(PlayList)의 데이터 베이스, 기록 매체(100)의 기록 내용의 관리 정보(info.dvr) 및 썸네일 화상의 정보를 작성한다. 이들 정보로 구성되는 어플리케이션 데이터 베이스 정보는 AV 스트림과 마찬가지로 하여, ECC 부호화부(20), 변조부(21)에서 처리되어, 기입부(22)에 입력된다. 기입부(22)는 제어부(23)로부터 출력되는 제어 신호에 기초하여, 기록 매체(100)에 데이터 베이스 파일을 기록한다.

<192> 상술한 어플리케이션 데이터 베이스 정보에 대한 상세는 후술한다.

<193> 이와 같이 하여 기록 매체(100)에 기록된 AV 스트림 파일(화상 데이터와 음성 데이터의 파일)과, 어플리케이션 데이터 베이스 정보가 재생부(3)에 의해 재생되는 경우, 우선, 제어부(23)는 판독부(28)에 대하여 기록 매체(100)로부터 어플리케이션 데이터 베이스 정보를 판독하도록 지시한다. 그리고, 판독부(28)는 기록 매체(100)로부터 어플리케이션 데이터 베이스 정보를 판독하고, 그 어플리케이션 데이터 베이스 정보는 복조부(29), ECC 복호부(30)의 복조와 오류 정정 처리를 거쳐, 제어부(23)에 입력된다.

<194> 제어부(23)는 어플리케이션 데이터 베이스 정보에 기초하여, 기록 매체(100)에 기록되어 있는 PlayList의 일람을 단자(24)의 사용자 인터페이스로 출력한다. 사용자는 PlayList의 일람으로부터 재생하고자 하는 PlayList를 선택하고, 재생이 지정된 PlayList에 관한 정보가 제어부(23)에 입력된다. 제어부(23)는 그 PlayList의 재생에 필요한 AV 스트림 파일의 판독을 판독부(28)에 지시한다. 판독부(28)는 그 지시에 따라, 기록 매체(100)로부터 대응하는 AV 스트림을 판독하여 복조부(29)로 출력한다. 복조부(29)에 입력된 AV 스트림은 소정의 처리가 실시됨으로써 복조되며, 다시 ECC 복호부(30)의 처리를 거쳐 소스 디패킷타이저(31)로 출력된다.

<195> 소스 디패킷타이저(31)는 기록 매체(100)로부터 판독되어 소정의 처리가 실시된 어플리케이션 포맷의 AV 스트림을 디멀티플렉서(26)가 처리 가능한 스트림으로 변환한다. 디멀티플렉서(26)는 제어부(23)에 의해 지정된 AV 스트림의 재생 구간(PlayItem)을 구성하는 비디오 스트림(V), 오디오 스트림(A) 및 AV 동기 등의 시스템 정보(S)를 AV 디코더(27)로 출력한다. AV 디코더(27)는 비디오 스트림과 오디오 스트림을 복호하고, 재생 비디오 신호와 재생 오디오 신호를 각각 대응하는 단자(32)와 단자(33)로부터 출력한다.

<196> 또한, 사용자 인터페이스로서의 단자(24)로부터 랜덤 액세스 재생이나 특수 재생을 지시하는 정보가 입력된 경우, 제어부(23)는 AV 스트림의 데이터 베이스(Clip)의 내용에 기초하여, 기억 매체(100)로부터의 AV 스트림의 판독 위치를 결정하고, 그 AV 스트림의 판독을 판독부(28)에 지시한다. 예를 들면, 사용자에 의해 선택된 PlayList를 소정의 시각으로부터 재생하는 경우, 제어부(23)는 지정된 시각에 가장 가까운 타임 스탬프를 갖는 I픽쳐로부터의 데이터를 판독하도록 판독부(28)에 지시한다.

<197> 또한, Clip Information 중의 ClipMark에 저장되어 있는 프로그램의 재생개시점이나 장면 전환점 중으로부터, 사용자가 임의의 클립마크를 선택했을 때(예를 들면, 이 동작은 ClipMark에 저장되어 있는 프로그램의 검색점이나 장면 전환점의 썸네일 화상 리스트를 사용자 인터페이스에 표시하여, 사용자가 그 중에서 임의의 화상을 선택함으로써 행해짐), 제어부(23)는 Clip Information의 내용에 기초하여 기록 매체(100)로부터 AV 스트림의 판독 위치를 결정하고, 그 AV 스트림의 판독을 판독부(28)로 지시한다. 즉, 사용자가 선택한 화상이 저장되어 있는 AV 스트림 상에서의 어드레스에 가장 가까운 어드레스에 임의의 I픽쳐로부터의 데이터를 판독하도록 판독부(28)로 지시한다. 판독부(28)는 지정된 어드레스로부터 데이터를 판독하여, 판독된 데이터는 복조부(29), ECC 복호부(30), 소스 패킷타이저(31)의 처리를 거쳐 멀티플렉서(26)로 입력되고, AV 디코더(27)에 의해 복호되어, 마크점의 픽쳐의 어드레스로 나타내는 AV 데이터가 재생된다.

<198> 또한, 사용자에 의해 고속 재생(Fast-forward playback)이 지시된 경우, 제어부(23)는 AV 스트림의 데이터 베이스(Clip)에 기초하여, AV 스트림 중의 I픽쳐 데이터를 순차적으로 연속하여 판독하도록 판독부(28)에 지시한다.

<199> 판독부(28)는 지정된 랜덤 액세스 포인트로부터 AV 스트림의 데이터를 판독하고, 판독된 데이터는 후단의 각 부의 처리를 거쳐 재생된다.

<200> 다음으로, 사용자가 기록 매체(100)에 기록되어 있는 AV 스트림의 편집을 행하는 경우를 설명한다. 사용자가 기록 매체(100)에 기록되어 있는 AV 스트림의 재생 구간을 지정하여 새로운 재생 경로를 작성하고자 하는 경우, 예를 들면, 프로그램 A라는 가요 프로그램으로부터 가수 A의 부분을 재생하고, 그 후 계속해서, 프로그램 B라는 가요 프로그램의 가수 A의 부분을 재생하고자 하는 재생 경로를 작성하고자 하는 경우, 사용자 인터페이스로서의 단자(24)로부터 재생 구간의 개시점(인점)과 종료점(아웃점)의 정보가 제어부(23)에 입력된다. 제어부(23)는 AV 스트림의 재생 구간(PlayItem)을 그룹화한 것(PlayList)의 데이터 베이스를 작성한다.

<201> 사용자가 기록 매체(100)에 기록되어 있는 AV 스트림의 일부를 소거하고자 하는 경우, 사용자 인터페이스로서의 단자(24)로부터 소거 구간의 인점과 아웃점의 정보가 제어부(23)에 입력된다. 제어부(23)는 필요한 AV 스트림 부분만을 참조하도록 PlayList의 데이터 베이스를 변경한다. 또한, AV 스트림이 불필요한 스트림 부분을 소거하도록 기입부(22)에 지시한다.

<202> 사용자가 기록 매체(100)에 기록되어 있는 AV 스트림의 재생 구간을 지정하여 새로운 재생 경로를 작성하고자 하는 경우에 대해, 또한, 각각의 재생 구간을 이음매 없이(심리스) 접속하고자 하는 경우에 대해 설명한다. 이러한 경우, 제어부(23)는 AV 스트림의 재생 구간(PlayItem)을 그룹화한 것(PlayList)의 데이터 베이스를 작성하고, 또한, 재생 구간의 접속점 부근의 비디오 스트림의 부분적인 재차 인코드와 재차 다중화를 행한다.

<203> 우선, 단자(24)로부터 재생 구간의 인점의 픽쳐의 정보와, 아웃점의 픽쳐의 정보가 제어부(23)로 입력된다. 제어부(23)는 판독부(28)에 인점측 픽쳐와 아웃점측의 픽쳐를 재생하기 위해 필요한 데이터의 판독을 지시한다. 그리고, 판독부(28)는 기록 매체(100)로부터 데이터를 판독하고, 그 데이터는 복조부(29), ECC 복호부(30), 소스 디패킷타이저(31)를 거쳐, 디멀티플렉서(26)로 출력된다.

<204> 제어부(23)는 디멀티플렉서(26)에 입력된 데이터를 해석하여, 비디오 스트림의 재차 인코드 방법(picture_coding_type의 변경, 재차 인코드하는 부호화 비트량의 할당)과, 재차 다중화 방식을 결정하고, 그 방식을 AV 인코더(15)와 멀티플렉서(16)에 공급한다.

<205> 다음으로, 디멀티플렉서(26)는 입력된 스트림을 비디오 스트림(V), 오디오 스트림(A) 및 시스템 정보(S)로 분리한다. 비디오 스트림은 AV 디코더(27)에 입력되는 데이터와 멀티플렉서(16)에 입력되는 데이터가 있다. 전자의 데이터는 재차 인코드하기 위해 필요한 데이터로, 이것은 AV 디코더(27)에서 복호되고, 복호된 픽쳐는 AV 인코더(15)에서 재차 인코드되어 비디오 스트림으로 된다. 후자의 데이터는 재차 인코드를 하지 않고, 오리지널 스트림으로부터 복사되는 데이터이다. 오디오 스트림, 시스템 정보에 대해서는 직접적으로 멀티플렉서(16)에 입력된다.

<206> 멀티플렉서(16)는, 제어부(23)로부터 입력된 정보에 기초하여, 입력 스트림을 다중화하여 다중화 스트림을 출력한다. 다중화 스트림은 ECC 부호화부(20), 변조부(21)에서 처리되어, 기입부(22)에 입력된다. 기입부(22)는 제어부(23)로부터 공급되는 제어 신호에 기초하여, 기록 매체(100)에 AV 스트림을 기록한다.

<207> 이하, 어플리케이션 데이터 베이스 정보나, 그 정보에 기초하는 재생, 편집의 조작에 관한 설명을 한다. 도 2는 어플리케이션 포맷의 구조를 설명하는 도면이다. 어플리케이션 포맷은 AV 스트림의 관리를 위해 PlayList와 Clip의 2개의 층을 갖는다. Volume Information은 디스크 내의 모든 Clip과 PlayList의 관리를 행한다. 여기서는, 하나의 AV 스트림과 그 부속 정보의 쌍을 하나의 오브젝트라고 생각하고, 이것을 Clip이라 한다. AV 스트림 파일은 Clip AV stream file이라 생각하고, 이 부속 정보는 Clip Information file이라 한다.

<208> 하나의 Clip AV Stream file는 MPEG2 전송 스트림을 어플리케이션 포맷에 의해 규정되는 구조로 배치한 데이터를 저장한다. 일반적으로, 파일은 바이트 열로서 처리되지만, Clip AV stream file의 콘텐츠는 시간축 상에 전개되고, Clip 중의 엔트리 포인트(I픽쳐)는 주로 시간 베이스로 지정된다. 소정의 Clip로의 액세스 포인트의 타임 스탬프가 제공되었을 때, Clip Information file는 Clip AV stream file 중에서 데이터의 판독을 개시하기 위한 어드레스 정보를 찾는데 도움이 된다.

<209> PlayList에 대하여, 도 3을 참조하여 설명한다. PlayList는 Clip 내에서 사용자가 보고자 하는 재생 구간을 선택하고, 이것을 간단하게 편집할 수 있도록 하기 위해 설치되어 있다. 하나의 PlayList는 Clip 내의 재생 구간의 집합이다. 소정의 Clip 내의 하나의 재생 구간은 PlayItem이라 하고, 그것은 시간축 상의 인점(IN)과 아웃점(OUT)의 쌍으로 나타낸다. 따라서, PlayList는 복수의 PlayItem이 모여서 구성된다.

<210> PlayList에는 2개의 타입이 있다. 하나는 Real PlayList이고, 다른 하나는 Virtual PlayList이다. Real PlayList는 그것이 참조하고 있는 Clip의 스트림 부분을 공유하고 있다. 즉, Real PlayList는 그것이 참조하고 있는 Clip의 스트림 부분에 상당하는 데이터 용량을 디스크 내에서 점유하고, Real PlayList가 소거된 경우, 그것이 참조하고 있는 Clip의 스트림 부분의 데이터도 소거된다.

<211> Virtual PlayList는 Clip의 데이터를 공유하지 않는다. 따라서, Virtual PlayList가 변경 또는 소거되었다고 해도, Clip의 내용에는 어떠한 변화도 생기지 않는다.

<212> 다음으로, Real PlayList의 편집에 대하여 설명한다. 도 4의 A는 Real PlayList의 작성(create)에 관한 도면으로, AV 스트림이 새로운 clip으로서 기록되는 경우, 그 Clip 전체를 참조하는 Real PlayList가 새롭게 작성되는 조작이다.

<213> 도 4의 B는 Real PlayList의 분할(divide)에 관한 도면으로, Real PlayList가 원하는 점에서 나누어져, 2개의 Real PlayList로 분할되는 조작이다. 이 분할 조작은, 예를 들면, 하나의 PlayList에 의해 관리되고 있는 하나의 클립 내에 2개의 프로그램이 관리되도록 한 경우에, 사용자가 각각의 프로그램으로서 다시 등록(기록)하고자 할 때에 행해진다. 이 조작에 의해, Clip의 내용이 변경되지는 않는다(Clip 자체가 분할되지는 않는다).

<214> 도 4의 C는 Real PlayList의 결합(combine)에 관한 도면으로, 2개의 Real PlayList를 결합하여, 하나의 새로운 Real PlayList로 하는 조작이다. 이 결합 조작은, 예를 들면, 사용자가 2개의 프로그램을 하나의 프로그램으로서 다시 등록하고자 할 때에 행해진다. 이 조작에 의해, Clip이 변경되지는 않는다(Clip 자체가 하나로 되지는 않는다).

<215> 도 5의 A는 Real PlayList 전체의 삭제(delete)에 관한 도면으로, 소정의 Real PlayList 전체를 소거하는 조작이 행해진 경우, 삭제된 Real PlayList가 참조하는 Clip의 대응하는 스트림 부분도 삭제된다.

<216> 도 5의 B는 Real PlayList의 부분적인 삭제에 관한 도면으로, Real PlayList의 원하는 부분이 삭제된 경우, 대응하는 PlayItem이 필요한 Clip의 스트림 부분만을 참조하도록 변경된다. 그리고, Clip의 대응하는 스트림 부분은 삭제된다.

<217> 도 5의 C는 Real PlayList의 최소화(Minimize)에 관한 도면으로, Real PlayList에 대응하는 PlayItem을 Virtual PlayList에 필요한 Clip의 스트림 부분만을 참조하도록 하는 조작이다. Virtual PlayList에 있어서 불필요한 Clip의 대응하는 스트림 부분은 삭제된다.

<218> 상술한 바와 같은 조작에 의해, Real PlayList가 변경되어, 그 Real PlayList가 참조하는 Clip의 스트림 부분이 삭제된 경우, 그 삭제된 Clip을 사용하고 있는 Virtual PlayList가 존재하고, 그 Virtual PlayList에서 삭제된 Clip에 의해 문제가 생길 가능성이 있다.

<219> 그와 같은 문제가 생기지 않도록, 사용자에게 삭제 조작에 대하여, 「그 Real PlayList가 참조하고 있는 Clip의 스트림 부분을 참조하고 있는 Virtual P layList가 존재하고, 만약 그 Real PlayList가 소거되면, 그 Virtual PlayList도 또한 소거되게 되지만, 그래도 괜찮습니까?」라는 메시지 등을 표시시킴으로써, 확인(경고)을 재촉한 후에, 사용자의 지시에 의해 삭제 처리를 실행하거나, 또는, 캔슬한다. 또는, Virtual PlayList를 삭제하는 대신에, Real PlayList에 대하여 최소화 조작이 행해지도록 한다.

<220> 다음으로 Virtual PlayList에 대한 조작에 대하여 설명한다. Virtual PlayList에 대하여 조작이 행해졌다고 해도, Clip의 내용이 변경되지는 않는다. 도 6의 A 및 B는 어셈블(Assemble) 편집(IN-OUT 편집)에 관한 도면으로, 사용자가 보고자 원하는 재생 구간의 PlayItem을 작성하고, Virtual PlayList를 작성하는 조작이다. PlayItem간의 심리스 접속이 어플리케이션 포맷에 의해 서포트되고 있다(후술).

<221> 도 6의 A에 도시한 바와 같이, 2개의 Real PlayList(1, 2)와, 각각의 Real Play List에 대응하는 Clip(1, 2)가 존재하고 있는 경우에, 사용자가 Real PlayList1 내의 소정 구간(In1 내지 Out1까지의 구간: PlayItem(1))을 재생 구간으로서 지시하고, 계속해서 재생하는 구간으로서 Real PlayList(2) 내의 소정 구간(In2 내지 Out2까지의 구간: PlayItem(2))을 재생 구간으로서 지시했을 때, 도 6의 B에 도시한 바와 같이, PlayItem(1)과 PlayItem(2)으로 구성되는 하나의 Virtual PlayList가 작성된다.

<222> 다음으로, Virtual PlayList의 재편집(Re-editing)에 대하여 설명한다. 재편집에는 Virtual PlayList 중의 인 점이나 아웃점의 변경, Virtual PlayList로의 새로운 PlayItem의 삽입(insert)이나 추가(append), Virtual PlayList 중의 PlayItem의 삭제 등이 있다. 또한, Virtual PlayList 그 자체를 삭제할 수도 있다.

<223> 도 7은 Virtual PlayList로의 오디오의 후기록(Audio dubbing(post recording))에 관한 도면으로, Virtual PlayList로의 오디오의 후기록을 서브 패스로서 등록하는 조작이다. 이 오디오의 후기록은 어플리케이션 포맷에 의해 서포트되고 있다. Virtual PlayList의 메인 패스의 AV 스트림에 부가적인 오디오 스트림이 서브 패스로서 부가된다.

<224> Real PlayList와 Virtual PlayList에서 공통의 조작으로서, 도 8에 도시한 바와 같은 PlayList의 재생 순서의 변경(Moving)이 있다. 이 조작은 디스크(볼륨) 내에서의 PlayList의 재생 순서의 변경으로, 어플리케이션 포맷에 있어서 정의되는 Table Of PlayList(도 20 등을 참조하여 후술함)에 의해 서포트된다. 이 조작에 의해, Clip의 내용이 변경되지는 않는다.

<225> 다음으로, 마크(Mark)에 대하여 설명한다. 마크는 Clip 및 PlayList 내의 하이라이트나 특징적인 시간을 지정하기 위해 부가되어 있다. Clip에 부가되는 마크는 AV 스트림의 내용에 기인하는 특징적인 장면을 지정하는, 예를 들면, 프로그램 검색점이나 장면 전환점 등이다. ClipMark는 도 1의 에를 들면 해석부(14)에 의해 생성된다. PlayList를 재생할 때, 그 PlayList가 참조하는 Clip의 마크를 참조하여, 사용할 수 있다.

<226> PlayList에 부가되는 마크는 PlayListMark(플레이 리스트 마크)라 불린다. PlayListMark는 주로 사용자에 의해 세트되는, 예를 들면, 북 마크나 리쥼점 등이다. Clip 또는 PlayList에 마크를 세트하는 것은 마크의 시각을 나타내는 타임 스탬프를 마크 리스트에 추가함으로써 행해진다. 또한, 마크를 삭제하는 것은 마크 리스트 중에서 그 마크의 타임 스탬프를 제거하는 것이다. 따라서, 마크의 설정이나 삭제에 의해, AV 스트림은 어떠한 변경도 되지 않는다.

<227> ClipMark의 다른 포맷으로서, ClipMark가 참조하는 픽처를 AV 스트림 중에서 어드레스 베이스로 지정하도록 하여도 된다. Clip에 마크를 세트하는 것은 마크점의 픽처를 나타내는 어드레스 베이스의 정보를 마크 리스트에 추가함으로써 행해진다. 또한, 마크를 삭제하는 것은 마크 리스트 중으로부터 그 마크점의 픽처를 나타내는 어드레스 베이스의 정보를 제거하는 것이다. 따라서, 마크의 지정이나 삭제에 의해 AV 스트림은 어떠한 것도 변경되지 않는다.

<228> 다음으로, 썸네일에 대하여 설명한다. 썸네일은 Volume, PlayList 및 Clip에 부가되는 정지 화상이다. 썸네일에는 2개의 종류가 있으며, 하나는 내용을 나타내는 대표 화상으로서의 썸네일이다. 이것은 주로 사용자가 커서(도시되지 않음) 등을 조작하여 보고자 하는 것을 선택하기 위한 메뉴 화면에서 사용되는 것이다. 다른 하나는 마크가 가리키고 있는 장면을 나타내는 화상이다.

<229> Volume과 각 PlayList는 대표 화상을 갖을 수 있도록 할 필요가 있다. Volume의 대표 화상은 디스크(기록 매체(100), 이하, 기록 매체(100)는 디스크 형상의 것으로 하고, 간단하게 디스크로 기술함)를 기록 재생 장치(1)의 소정의 장소에 세트하였을 때에, 그 디스크의 내용을 나타내는 정지 화상을 최초로 표시하는 경우 등에 이용되

는 것을 상정하고 있다. PlayList의 대표 화상은 PlayList를 선택하는 메뉴 화면에서, PlayList의 내용을 나타내기 위한 정지 화상으로서 이용되는 것을 상정하고 있다.

<230> PlayList의 대표 화상으로서, PlayList의 최초의 화상을 썸네일(대표 화상)로 하는 것이 고려되지만, 반드시 재생 시각 0의 선두의 화상이 내용을 나타내는 데에 있어서 최적의 화상이라고는 할 수 없다. 그래서, PlayList의 썸네일로서, 임의의 화상을 사용자가 설정할 수 있도록 한다. 이상 volume을 나타내는 대표 화상으로서의 썸네일과, PlayList를 나타내는 대표 화상으로서의 썸네일의 2종류의 썸네일을 메뉴 썸네일이라 한다. 메뉴 썸네일은 빈번하게 표시되기 때문에, 디스크로부터 고속으로 판독될 필요가 있다. 이 때문에, 모든 메뉴 썸네일을 하나의 파일에 저장하는 것이 효율적이다. 메뉴 썸네일은 반드시 볼륨 내의 동화상으로부터 추출한 픽쳐일 필요는 없고, 도 10에 도시한 바와 같이, 퍼스널 컴퓨터나 디지털 스틸 카메라로부터 수신한 화상이라도 무방하다.

<231> 한편, Clip과 PlayList에는 여러개의 마크를 붙일 필요가 있고, 마크 위치의 내용을 알기 위해 마크점의 화상을 용이하게 볼 수 있도록 할 필요가 있다. 이러한 마크점을 나타내는 픽처를 마크 썸네일(Mark Thumbnails)이라 한다. 따라서, 썸네일의 기초가 되는 화상은 외부로부터 수신한 화상보다 마크점의 화상을 추출한 것이 주가 된다.

<232> 도 11은 PlayList에 붙여지는 마크와, 그 마크 썸네일의 관계에 대하여 나타내는 도면이고, 도 12는 Clip에 붙여지는 마크와, 그 마크 썸네일의 관계에 대하여 나타내는 도면이다. 마크 썸네일은 메뉴 썸네일과 달리 PlayList의 상세를 나타낼 때에 서브 메뉴 등으로 사용되기 때문에, 짧은 액세스 시간에서 판독되는 것이 요구되지 않는다. 그 때문에, 썸네일이 필요하게 될 때마다, 기록 재생 장치(1)가 파일을 개방하고, 그 파일의 일부를 판독함으로써 다소 시간이 걸리더라도 문제가 되지는 않는다.

<233> 또한, 볼륨 내에 존재하는 파일 수를 줄이기 위해, 모든 마크 썸네일은 하나의 파일에 저장하는 것이 좋다. PlayList는 메뉴 썸네일 1개와 복수의 마크 썸네일을 갖을 수 있지만, Clip은 직접 사용자가 선택할 필요성이 없기(통상, PlayList 경유로 지정하기) 때문에, 메뉴 썸네일을 마련할 필요는 없다.

<234> 도 13은 상술한 것을 고려한 경우의 메뉴 썸네일, 마크 썸네일, PlayList 및 Clip의 관계에 대하여 나타낸 도면이다. 메뉴 썸네일 파일에는 PlayList마다 마련된 메뉴 썸네일이 파일로 되어 있다. 메뉴 썸네일 파일에는 디스크에 기록되어 있는 데이터의 내용을 대표하는 볼륨 썸네일이 포함되어 있다. 마크 썸네일 파일은 각 PlayList마다와 각 Clip마다 작성된 썸네일이 파일로 되어 있다.

<235> 다음으로, CPI(Characteristic Point Information)에 대하여 설명한다. CPI는 Clip 인포메이션 파일에 포함되는 데이터로, 주로 그것은 Clip으로의 액세스 포인트의 타임 스탬프가 제공되었을 때, Clip AV stream file 중에서 데이터의 판독을 개시해야 할 데이터 어드레스를 찾아내기 위해 이용된다. 본 실시예에서는 2종류의 CPI를 이용한다. 하나는 EP_map이고, 다른 하나는 TU_map이다.

<236> EP_map은 엔트리 포인트(EP) 데이터의 리스트로, 이것은 엘리먼트리 스트림 및 전송 스트림으로부터 추출된 것이다. 이것은 AV 스트림 중에서 디코드를 개시해야 할 엔트리 포인트의 장소를 찾아내기 위한 어드레스 정보를 갖는다. 하나의 EP 데이터는 프레젠테이션 타임 스탬프(PTS)와, 그 PTS에 대응하는 액세스 유닛의 AV 스트림 중의 데이터 어드레스의 쌍으로 구성된다.

<237> EP_map은 주로 2개의 목적을 위해 사용된다. 첫번째는 PlayList 중에서 프레젠테이션 타임 스탬프에 의해 참조되는 액세스 유닛의 AV 스트림 중의 데이터 어드레스를 찾아내기 위해 사용된다. 두번째는 제1 포워드 재생이나 제1 리버스 재생를 위해 사용된다. 기록 재생 장치(1)가 입력 AV 스트림을 기록하는 경우, 그 스트림의 신택스를 해석할 수 있을 때, EP_map이 작성되어 디스크에 기록된다.

<238> TU_map은 디지털 인터페이스를 통해 입력되는 전송 패킷의 도착 시각에 기초한 타임 유닛(TU) 데이터의 리스트를 갖는다. 이것은 도착 시각 베이스의 시간과 AV 스트림 중의 데이터 어드레스와의 관계를 제공한다. 기록 재생 장치(1)가 입력 AV 스트림을 기록하는 경우, 그 스트림의 신택스를 해석할 수 없을 때, TU_map이 작성되어 디스크에 기록된다.

<239> STCInfo는 MPEG2 전송 스트림을 저장하고 있는 AV 스트림 파일 중에 있는 STC의 불연속점 정보를 저장한다. 만일, AV 스트림이 STC의 불연속점을 갖는 경우, 그 AV 스트림 파일 중에서 동일한 값의 PTS가 나타날 가능성이 있다. 그 때문에, AV 스트림 상의 소정의 시각을 PTS 베이스로 가리킬 때, 액세스 포인트의 PTS만으로는 그 포인트를 특정하기에는 불충분하다.

<240> 또한, 그 PTS를 포함하는 연속된 STC 구간의 인덱스가 필요하다. 연속된 STC 구간을 이 포맷에서는 STC-sequence라 하고, 그 인덱스를 STC-sequence-id라 한다. STC-sequence의 정보는 Clip Information file의 STCInfo로 정의된다. STC-sequence-id는 EP_map을 갖는 AV 스트림 파일로 사용하는 것으로, TU_map을 갖는 AV 스트림 파일에서는 옵션이다.

<241> 프로그램은 엘리먼트리 스트림의 집합으로, 이들 스트림의 동기 재생을 위해, 단지 하나의 시스템 타임 베이스를 공유하는 것이다. 재생 장치(도 1의 기록 재생 장치(1))에서, AV 스트림의 디코드에 앞서서, 그 AV 스트림의 내용을 아는 것은 유용하다. 예를 들면, 비디오나 오디오의 엘리먼트리 스트림을 전송하는 전송 패킷의 PID의 값이나, 비디오나 오디오의 컴포넌트 종류(예를 들면, HDTV의 비디오와 MPEG-2 AAC의 오디오 스트림 등) 등의 정보이다.

<242> 이 정보는 AV 스트림을 참조하는 PlayList의 내용을 사용자에게 설명하는 메뉴 화면을 작성하는 데 유용하고, 또한, AV 스트림의 디코드에 앞서서, 재생 장치의 AV 디코더 및 디멀티플렉서의 초기 상태를 세트하기 위해 도움이 된다. 이 이유 때문에, Clip Information file는 프로그램의 내용을 설명하기 위한 ProgramInfo를 갖는다.

<243> MPEG2 전송 스트림을 저장하고 있는 AV 스트림 파일은 파일 내에서 프로그램 내용이 변화될지도 모른다. 예를 들면, 비디오 엘리먼트리 스트림을 전송하는 전송 패킷의 PID가 변화되거나, 비디오 스트림의 컴포넌트의 종류가 SDTV로부터 HDTV로 변화되는 것 등이다.

<244> ProgramInfo는 AV 스트림 파일 내에서의 프로그램 내용의 변화점의 정보를 저장한다. AV 스트림 파일 내에서 이 포맷으로 결정되는 프로그램 내용이 일정한 구간을 Program-sequence라고 한다. Program-sequence는 EP_map을 갖는 AV 스트림 파일로 사용하는 것으로, TU_map을 갖는 AV 스트림 파일에서는 옵션이다.

<245> 본 실시예에서는, 셀프 인코드의 스트림 포맷(SESF)을 정의한다. SESF는 아날로그 입력 신호를 부호화하는 목적 및 디지털 입력 신호(예를 들면 DV)를 디코드하고 나서 MPEG2 전송 스트림으로 부호화하는 경우에 이용된다.

<246> SESF는 MPEG-2 전송 스트림 및 AV 스트림에 대한 엘리먼트리 스트림의 부호화 제한을 정의한다. 기록 재생 장치(1)가 SESF 스트림을 인코드하여, 기록하는 경우, EP_map이 작성되어 디스크에 기록된다.

<247> 디지털 방송의 스트림은 다음에 나타내는 방식 중 어느 하나가 이용되어 기록 매체(100)에 기록된다. 우선, 디지털 방송의 스트림을 SESF 스트림으로 변환 코딩한다. 이 경우, 기록된 스트림은 SESF에 준거해야 한다. 이 경우, EP_map이 작성되어 디스크에 기록되어야 한다.

<248> 혹은, 디지털 방송 스트림을 구성하는 엘리먼트리 스트림을 새로운 엘리먼트리 스트림으로 변환 코딩하고, 그 디지털 방송 스트림의 규격화 조직이 정하는 스트림포맷에 준거한 새로운 전송 스트림으로 재차 다중화한다. 이 경우, EP_map이 작성되어 디스크에 기록되어야 한다.

<249> 예를 들면, 입력 스트림이 ISDB(일본의 디지털 BS 방송의 규격 명칭) 준거의 MPEG-2 전송 스트림이고, 그것이 HDTV 비디오 스트림과 MPEG AAC 오디오 스트림을 포함하게 한다. HDTV 비디오 스트림을 SDTV 비디오 스트림으로 변환 코딩하고, 그 SDTV 비디오 스트림과 오리지널의 AAC 오디오 스트림을 TS로 재차 다중화한다. SDTV 스트림으로 기록되는 전송 스트림은 모두 ISDB 포맷에 준거해야 한다.

<250> 디지털 방송의 스트림이 기록 매체(100)에 기록될 때의 다른 방식으로서, 입력 전송 스트림을 트랜스페어런트하게 기록하는(입력 전송 스트림을 전혀 변경시키지 않고 기록하는) 경우로, 그 때에 EP_map이 작성되어 디스크에 기록된다.

<251> 또는, 입력 전송 스트림을 트랜스페어런트하게 기록하는(입력 전송 스트림을 전혀 변경시키지 않고 기록하는) 경우로, 그 때에 TU_map이 작성되어 디스크에 기록된다.

<252> 다음으로 디렉토리와 파일에 대하여 설명한다. 이하, 기록 재생 장치(1)를 DVR(Digital Video Recording)로 간단하게 기술한다. 도 14는 디스크 상의 디렉토리 구조의 일례를 나타내는 도면이다. DVR의 디스크 상에 필요한 디렉토리는, 도 14에 도시한 바와 같이, "DVR" 디렉토리를 포함하는 root 디렉토리, "PLAYLIST" 디렉토리, "CLIPINF" 디렉토리, "M2TS" 디렉토리 및 "DATA" 디렉토리를 포함하는 "DVR" 디렉토리이다. root 디렉토리의 아래에 이들 이외의 디렉토리를 작성하도록 해도 되지만, 이들은 본 실시예의 어플리케이션 포맷에서는 무시되도록 한다.

<253> "DVR" 디렉토리 아래에는 DVR 어플리케이션 포맷에 의해 규정되는 모든 파일과 디렉토리가 저장된다. "DVR" 디렉토리는 4개의 디렉토리를 포함한다. "PLAYLIST" 디렉토리 아래에는 Real PlayList와 Virtual PlayList의 데이터 베이스 파일이 놓인다. 이 디렉토리는 PlayList가 하나도 없어도 존재한다.

<254> "CLIPINF" 디렉토리 아래에는 Clip의 데이터 베이스가 놓인다. 이 디렉토리도 Clip이 하나도 없어도 존재한다. "M2TS" 디렉토리 아래에는 AV 스트림 파일이 놓인다. 이 디렉토리는 AV 스트림 파일이 하나도 없어도 존재한다. "DATA" 디렉토리는 디지털 TV 방송 등의 데이터 방송의 파일이 저장된다.

<255> "DVR" 디렉토리는 다음에 나타내는 파일을 저장한다. "info.dvr" 파일은 DVR 디렉토리 아래에 작성되고, 어플리케이션층의 전체적인 정보를 저장한다. DVR 디렉토리 아래에는 단 하나의 info.dvr이 있어야만 한다. 파일명은 info.dvr로 고정되도록 한다. "menu.thmb" 파일은 메뉴 썸네일 화상에 관련되는 정보를 저장한다. DVR 디렉토리 아래에는 0 또는 하나의 메뉴 썸네일이 있어야만 한다. 파일명은 memu.thmb로 고정되도록 한다. 메뉴 썸네일 화상이 하나도 없는 경우, 이 파일은 존재하지 않아도 된다.

<256> "mark.thmb" 파일은 마크 썸네일 화상에 관련되는 정보를 저장한다. DVR 디렉토리 아래에는 0 또는 하나의 마크 썸네일이 있어야만 한다. 파일명은 mark.thmb로 고정되도록 한다. 메뉴 썸네일 화상이 하나도 없는 경우, 이 파일은 존재하지 않아도 된다.

<257> "PLAYLIST" 디렉토리는 2종류의 PlayList 파일을 저장하는 것으로, 그들은 Real PlayList와 Virtual PlayList이다. "xxxxx.rpls" 파일은 하나의 Real PlayList에 관련되는 정보를 저장한다. 각각의 Real PlayList마다 하나의 파일이 작성된다. 파일명은 "xxxxx.rpls"이다. 여기서, "xxxxx"는 5개의 0 내지 9까지의 숫자이다. 파일 확장자는 "rpls"이어야 한다.

<258> "yyyyy.vpls" 파일은 하나의 Virtual PlayList에 관련되는 정보를 저장한다. 각각의 Virtual PlayList마다 하나의 파일이 작성된다. 파일명은 "yyyyy.vpls"이다. 여기서, "yyyyy"는 5개의 0 내지 9까지의 숫자이다. 파일 확장자는 "vpls"이어야 한다.

<259> "CLIPINF" 디렉토리는 각각의 AV 스트림 파일에 대응하여, 1개의 파일을 저장한다. "zzzzz.dip" 파일은 하나의 AV 스트림 파일(Clip AV stream file 또는 Bridge-Clip AV stream file)에 대응하는 Clip Information file이다. 파일명은 "zzzzz.dip"이고, "zzzzz"는 5개의 0 내지 9까지의 숫자이다. 파일 확장자는 "dip"이어야 한다.

<260> "M2TS" 디렉토리는 AV 스트림의 파일을 저장한다. "zzzzz.m2ts" 파일은 DVR 시스템에 의해 처리되는 AV 스트림 파일이다. 이것은 Clip AV stream file 또는 Bridge-Clip AV stream이다. 파일명은 "zzzzz.m2ts"이고, "zzzzz"는 5개의 0 내지 9까지의 숫자이다. 파일 확장자는 "m2ts"이어야 한다.

<261> "DATA" 디렉토리는 데이터 방송으로부터 전송되는 데이터를 저장하는 것이고, 데이터란, 예를 들면, XML file나 MHEG 파일 등이다.

<262> 다음으로, 각 디렉토리(파일)의 신택스와 시맨틱스를 설명한다. 우선, "info.dvr" 파일에 대하여 설명한다. 도 15는 "info.dvr" 파일의 신택스를 나타내는 도면이다. "info.dvr" 파일은 3개의 오브젝트로 구성되고, 그들은 DVRVolume(), TableOfPlayLists() 및 MakerPrivateData()이다.

<263> 도 15에 도시한 info.dvr의 신택스에 대하여 설명하면, TableOfPlayLists_Start_address는 info.dvr 파일의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, TableOfPlayList()의 선두 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다.

<264> MakerPrivateData_Start_address는 info.dvr 파일의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, MakerPrivateData()의 선두 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다. padding_word(패딩 워드)는 info.dvr의 신택스에 따라 삽입된다. N1과 N2는 0 또는 임의의 양의 정수이다. 각각의 패딩 워드는 임의의 값을 취하도록 해도 된다.

<265> DVRVolume()는 볼륨(디스크)의 내용을 기술하는 정보를 저장한다. 도 16은 DVRVolume()의 신택스를 나타내는 도면이다. 도 16에 도시한 DVR Volume()의 신택스를 설명하면, version_number는 이 DVRVolume()의 버전 번호를 나타내는 4개의 캐릭터 문자를 나타낸다. version_number는 ISO646에 따라 "0045"로 부호화된다.

<266> length는 이 length 필드의 직후부터 DVRVolume()의 최후까지의 DVRVolume()의 바이트 수를 나타내는 32비트의 부호없는 정수로 나타낸다.

<267> ResumeVolume()는 볼륨 중에서 마지막으로 재생한 Real PlayList 또는 Virtual PlayList의 파일명을 기억하고 있다. 단, Real PlayList 또는 Virtual PlayList의 재생을 사용자가 중단했을 때의 재생 위치는 PlayListMark()에 있어서 정의되는 resume-mark에 저장된다.

<268> 도 17은 ResumeVolume()의 신택스를 나타내는 도면이다. 도 17에 도시한 ResumeVolume()의 신택스를 설명하면, valid_flag는, 이 1비트의 플래그가 1로 세트되어 있는 경우, resume_PlayList_name 필드가 유효인 것을 나타내고, 이 플래그가 0으로 세트되어 있는 경우, resume_PlayList_name 필드가 무효인 것을 나타낸다.

<269> resume_PlayList_name의 10바이트의 필드는 리즘되어야 하는 Real PlayList 또는 Virtual PlayList의 파일명을 나타낸다.

<270> 도 16에 도시한 DVRVolume()의 신택스 내의 UIAppInfoVolume은 볼륨에 대한 사용자 인터페이스 어플리케이션의 파라미터를 저장한다. 도 18은 UIAppInfoVolume의 신택스를 나타내는 도면으로, 그 시맨틱스를 설명하면, character_set의 8비트의 필드는 Volume_name 필드로 부호화되어 있는 캐릭터 문자의 부호화 방법을 나타낸다. 그 부호화 방법은 도 19에 도시한 값에 대응한다.

<271> name_length의 8비트 필드는 Volume_name 필드 중에 나타내는 볼륨명의 바이트 길이를 나타낸다. Volume_name의 필드는 볼륨의 명칭을 나타낸다. 이 필드 중의 좌측으로부터 name_length수의 바이트 수가 유효한 캐릭터 문자로, 그것은 볼륨의 명칭을 나타낸다. Volume_name 필드 중에서, 이들 유효한 캐릭터 문자 후의 값은 어떤 값이 들어가도 된다.

<272> Volume_protect_flag는 볼륨 중의 콘텐츠를 사용자에게 제한하지 않고 보여주는 것이 좋은지의 여부를 나타내는 플래그이다. 이 플래그가 1로 세트되어 있는 경우, 사용자가 정확하게 PIN 번호(패스워드)를 입력할 수 있었을 때에만, 그 볼륨의 콘텐츠를 사용자에게 보여주는 것(재생되는 것)이 허가된다. 이 플래그가 0으로 세트되어 있는 경우, 사용자가 PIN 번호를 입력하지 않아도, 그 볼륨의 콘텐츠를 사용자에게 보여주는 것이 허가된다.

<273> 처음에, 사용자가 디스크를 플레이어에 삽입한 시점에 있어서, 혹시 이 플래그가 0으로 세트되어 있거나, 또는, 이 플래그가 1로 세트되어 있어도 사용자가 PIN 번호를 정확하게 입력할 수 있다면, 기록 재생 장치(1)는 그 디스크 중의 PlayList의 일람을 표시시킨다. 각각의 PlayList의 재생 제한은 volume_protect_flag와는 무관계하며, 그것은 UIAppInfoPlayList() 중에 정의되는 playback_control_flag에 의해 나타낸다.

<274> PIN은 4개의 0 내지 9까지의 숫자로 구성되고, 각각의 숫자는 ISO/IEC646에 따라 부호화된다. ref_thumbnail_index의 필드는 볼륨에 부가되는 썸네일 화상의 정보를 나타낸다. ref_thumbnail_index 필드가 OxFFFF가 아닌 값인 경우, 그 볼륨에는 썸네일 화상이 부가되어 있고, 그 썸네일 화상은 menu.thum 파일 중에 저장되어 있다. 그 화상은 menu.thum 파일 중에서 ref_thumbnail_index의 값을 이용하여 참조된다. ref_thumbnail_index 필드가 OxFFFF인 경우, 그 볼륨에는 썸네일 화상이 부가되어 있지 않은 것을 나타낸다.

<275> 다음으로, 도 15에 도시한 info.dvr의 신택스 내의 TableOfPlayLists()에 대하여 설명한다. TableOfPlayLists()는 PlayList(Real PlayList와 Virtual PlayList)의 파일명을 저장한다. 볼륨에 기록되어 있는 모든 PlayList 파일은 TableOfPlayList() 중에 포함된다. TableOfPlayLists()는 볼륨 중의 PlayList의 디폴트의 재생 순서를 나타낸다.

<276> 도 20은 TableOfPlayLists()의 신택스를 나타내는 도면으로, 그 신택스에 대하여 설명하면, TableOfPlayLists의 version_number는 이 TableOfplayLists의 버전 번호를 나타내는 4개의 캐릭터 문자를 나타낸다. version_number는 ISO646에 따라 "0045"로 부호화되어야 한다.

<277> length는 이 length 필드의 직후부터 TableOfPlayLists()의 최후까지의 TableOfPlayLists()의 바이트 수를 나타내는 32비트의 부호없는 정수이다. number_of_PlayLists의 16비트의 필드는 PlayLisLfile_name를 포함하는 for_loop의 루프 횟수를 나타낸다. 이 숫자는 볼륨에 기록되어 있는 PlayList의 수와 동일해야 한다. PlayList_file_name의 10바이트의 숫자는 PlayList의 파일명을 나타낸다.

<278> 도 21은 TableOfPlayLists()의 신택스의 다른 예의 구성을 나타내는 도면이다. 도 21에 도시한 신택스는 도 20에 도시한 신택스에 UIAppinfoPlayList(후술)를 포함시킨 구성으로 되어 있다. 이와 같이, UIAppinfoPlayList를 포함시킨 구성으로 함으로써, TableOfPlayLists를 판독하는 것만으로 메뉴 화면을 작성하는 것이 가능해진다. 여기서는, 도 20에 도시한 신택스를 이용하는 것으로 하여 이하의 설명을 한다.

<279> 도 15에 도시한 info.dvr의 신택스 내의 MakersPrivateData에 대하여 설명한다. MakersPrivateData는 기록 재

생 장치(1)의 메이커가 각 사의 특별한 어플리케이션을 위해, MakersPrivateData() 내에 메이커의 프라이비트 데이터를 삽입할 수 있도록 마련되어 있다. 각 메이커의 프라이비트 데이터는 이것을 정의한 메이커를 식별하기 위해 표준화된 maker_ID를 갖는다. MakersPrivateData()는 1개 이상의 maker_ID를 포함해도 된다.

<280> 소정의 메이커가 프라이비트 데이터를 삽입하고자 할 때에, 이미 다른 메이커의 프라이비트 데이터가 MakersPrivateData()에 포함되어 있는 경우, 다른 메이커는 이미 있는 오래된 프라이비트 데이터를 소거하는 것은 아니라, 새로운 프라이비트 데이터를 MakersPrivateData() 내에 추가하도록 한다. 이와 같이, 본 실시예에서는, 복수의 메이커의 프라이비트 데이터가 하나의 MakersPrivateData()에 포함되는 것이 가능하도록 한다.

<281> 도 22는 MakersPrivateData의 신택스를 나타내는 도면이다. 도 22에 도시한 MakersPrivateData의 신택스에 대하여 설명하면, version_number는 이 MakersPrivateData()의 버전 번호를 나타내는 4개의 캐릭터 문자를 나타낸다. version_number는 ISO646에 따라 "0045"로 부호화되어야 한다. length는 이 length 필드의 직후부터 MakersPrivateData()의 최후까지의 MakersPrivateData()의 바이트 수를 나타내는 32비트의 부호없는 정수를 나타낸다.

<282> mpd_blocks_start_address는 MakersPrivateData()의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, 최초의 mpd_block()의 선두 바이트 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다. number_of_maker_entries는 MakersPrivateData() 내에 포함되어 있는 메이커 프라이비트 데이터의 엔트리 수를 제공하는 16비트의 부호없는 정수이다. MakersPrivateData() 내에 동일한 maker_ID의 값을 갖는 메이커 프라이비트 데이터가 2개 이상 존재해서는 안된다.

<283> mpd_block_size는 1024바이트를 단위로 하여, 하나의 mpd_block의 크기를 제공하는 16비트의 부호없는 정수이다. 예를 들면, mpd_block_size=1이면, 그것은 하나의 mpd_block의 크기가 1024바이트인 것을 나타낸다. number_of_mpd_blocks는 MakersPrivateData() 내에 포함되는 mpd_block의 수를 제공하는 16비트의 부호없는 정수이다. maker_ID는 그 메이커 프라이비트 데이터를 작성한 DVR 시스템의 제조 메이커를 나타내는 16비트의 부호없는 정수이다. maker_ID로 부호화되는 값은 이 DVR 포맷의 라이센서에 의해 지정된다.

<284> maker_model_code는 그 메이커 프라이비트 데이터를 작성한 DVR 시스템의 모델 번호 코드를 나타내는 16비트의 부호없는 정수이다. maker_model_code로 부호화되는 값은 이 포맷의 라이센스를 받은 제조 메이커에 의해 설정된다. start_mpd_block_number는 그 메이커 프라이비트 데이터가 개시되는 mpd_block의 번호를 나타내는 16비트의 부호없는 정수이다. 메이커 프라이비트 데이터의 선두 데이터는 mpd_block의 선두에 위치하여야 한다. start_mpd_block_number는 mpd_block의 for-loop 내의 변수 j에 대응한다.

<285> mpd_length는 바이트 단위로 메이커 프라이비트 데이터의 크기를 나타내는 32비트의 부호없는 정수이다. mpd_block는 메이커 프라이비트 데이터가 저장되는 영역이다. MakersPrivateData() 내의 모든 mpd_block는 동일한 사이즈이어야 한다.

<286> 다음으로, Real PlayList file와 Virtual PlayList file에 대하여, 바꾸어 말하면, xxxxx.rpls와 yyyyy.vpls에 대하여 설명한다. 도 23은 xxxxx.rpls(Real PlayList), 또는, yyyyy.vpls(Virtual PlayList)의 신택스를 나타내는 도면이다. xxxxx.rpls와 yyyyy.vpls는 동일한 신택스 구성을 갖는다. xxxxx.rpls와 yyyyy.vpls는 각각 3개의 오브젝트로 구성되고, 그들은 PlayList(), PlayListMark() 및 MakerPrivateData()이다.

<287> PlayListMark_Start_address는 PlayList 파일의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, PlayListMark()의 선두 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다.

<288> MakerPrivateData_Start_address는 PlayList 파일의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, MakerPrivateData()의 선두 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다.

<289> padding_word(패딩 워드)는 PlayList 파일의 신택스에 따라 삽입되고, N1과 N2는 0 또는 임의의 양의 정수이다. 각각의 패딩 워드는 임의의 값을 취하도록 해도 된다.

<290> 여기서, 이미 간단하게 설명하였지만, PlayList에 대하여 다시 설명한다. 디스크 내에 있는 모든 Real PlayList에 의해 Bridge-Clip(후술)을 제외한 모든 Clip 내의 재생 구간이 참조되어야 한다. 또한, 2개 이상의 Real Play List가 이들 PlayItem으로 나타내는 재생 구간을 동일한 Clip 내에서 오버랩시켜서는 안된다.

<291> 도 24의 A 내지 도 24의 C를 참조하여 더 설명하면, 도 24의 A에 도시한 바와 같이, 모든 Clip은 대응하는 RealPlayList가 존재한다. 이 규칙은, 도 24의 B에 도시한 바와 같이, 편집 작업이 행해진 후에 있어서도 지켜

진다. 따라서, 모든 Clip은 어느 하나의 Real PlayList를 참조함으로써, 반드시 시청하는 것이 가능하다.

<292> 도 24의 C에 도시한 바와 같이, Virtual PlayList의 재생 구간은 Real PlayList의 재생 구간 또는 Bridge-Clip의 재생 구간 중에 포함되어 있어야 한다. 어떤 Virtual PlayList에도 참조되지 않는 Bridge-Clip이 디스크 중에 존재해서는 안된다.

<293> RealPlayList는 PlayItem의 리스트를 포함하지만, SubPlayItem을 포함해서는 안된다. Virtual PlayList는 PlayItem의 리스트를 포함하고, PlayList() 중에 도시되는 CPI_type이 EP_map type이고, 또한 PlayList_type이 0(비디오와 오디오를 포함하는 PlayList)인 경우, Virtual PlayList는 하나의 SubPlayItem을 포함할 수 있다. 본 실시예에서의 PlayList()에서는 SubPlayItem은 오디오의 후기록의 목적만으로 사용되고, 그리고, 하나의 Virtual PlayList가 갖는 SubPlayItem의 수는 0 또는 1이어야 한다.

<294> 다음으로, PlayList에 대하여 설명한다. 도 25는 PlayList의 신택스를 나타내는 도면이다. 도 25에 도시한 PlayList의 신택스를 설명하면, version_number는 이 PlayList()의 버전 번호를 나타내는 4개의 캐릭터 문자이다. version_number는 ISO646에 따라 "0045"로 부호화되어야 한다. length는 이 length 필드의 직후부터 PlayList()의 최후까지의 PlayList()의 바이트 수를 나타내는 32비트의 부호없는 정수이다. PlayList_type은 이 PlayList의 타입을 나타내는 8비트의 필드이고, 그 일례를 도 26에 도시한다.

<295> CPI_type은 1비트의 플래그로, PlayItem() 및 SubPlayItem()에 의해 참조되는 Clip의 CPI_type의 값을 나타낸다. 하나의 PlayList에 의해 참조되는 모든 Clip은 이들의 CPI() 중에 정의되는 CPI_type의 값이 동일해야만 한다. number_of_PayItems는 PlayList 중에 있는 PlayItem의 수를 나타내는 16비트의 필드이다.

<296> 소정의 PlayItem()에 대응하는 PlayItem_id는 PlayItem()을 포함하는 for-loop 중에서 그 PlayItem()이 나타나는 순서에 의해 정의된다. PlayItem_id는 0부터 개시된다. number_of_SubPlayItems는 PlayList 중에 있는 SubPlayItem의 수를 나타내는 16비트의 필드이다. 이 값은 0 또는 1이다. 부가적인 오디오 스트림의 패스(오디오 스트림 패스)는 서브 패스의 일종이다.

<297> 다음으로, 도 25에 도시한 PlayList의 신택스의 UIAppInfoPlayList에 대하여 설명한다. UIAppInfoPlayList는 PlayList에 대한 사용자 인터페이스 어플리케이션의 파라미터를 저장한다. 도 27은 UIAppInfoPlayList의 신택스를 나타내는 도면이다. 도 27에 도시한 UIAppInfoPlayList의 신택스를 설명하는 데 있어서, character_set는 8비트의 필드로, PlayList_name 필드에 부호화되어 있는 캐릭터 문자의 부호화 방법을 나타낸다. 그 부호화 방법은 도 19에 도시한 테이블에 준거하는 값에 대응한다.

<298> name_length는 8비트 필드로, PlayList_name 필드 중에 나타내는 PlayList명의 바이트 길이를 나타낸다. PlayList_name의 필드는 PlayList의 명칭을 나타낸다. 이 필드 중의 좌측으로부터 name_length수의 바이트 수가 유효한 캐릭터 문자로, 그것은 PlayList의 명칭을 나타낸다. PlayList_name 필드 중에서, 이들 유효한 캐릭터 문자 뒤의 값은 어떤 값이 들어가도 무방하다.

<299> record_time_and_date는 PlayList가 기록되었을 때의 일시를 저장하는 56비트의 필드이다. 이 필드는 년/월/일/시/분/초에 대하여, 14개의 숫자를 4비트의 Binary Coded Decimal(BCD)로 부호화한 것이다. 예를 들면, 2001/12/23:01:02:03은 "0x20011223010203"으로 부호화된다.

<300> duration은 PlayList의 총 재생 시간을 시간/분/초의 단위로 나타낸 24비트의 필드이다. 이 필드는 6개의 숫자를 4비트의 Binary Coded Decimal(BCD)로 부호화한 것이다. 예를 들면, 01:45:30은 "0x014530"으로 부호화된다.

<301> valid_period는 PlayList가 유효인 기간을 나타내는 32비트의 필드이다. 이 필드는 8개의 숫자를 4비트의 Binary Coded Decimal(BCD)로 부호화한 것이다. 예를 들면, 기록 재생 장치(1)는 이 유효 기간이 지난 PlayList를 자동 소거하도록 이용된다. 예를 들면, 2001/05/07은 "0x20010507"로 부호화된다.

<302> maker_id는 그 PlayList를 마지막으로 갱신한 DVR 플레이어(기록 재생 장치(1))의 제조자를 나타내는 16비트의 부호없는 정수이다. maker_id로 부호화되는 값은 DVR 포맷의 라이센서에 의해 할당된다. maker_code는 그 PlayList를 마지막으로 갱신한 DVR 플레이어의 모델 번호를 나타내는 16비트의 부호없는 정수이다. maker_code로 부호화되는 값은 DVR 포맷의 라이센스를 받은 제조자에 의해 결정된다.

<303> playback_control_flag의 플래그가 1로 세트되어 있는 경우, 사용자가 정확하게 PIN 번호를 입력할 수 있는 경우에만 그 PlayList는 재생된다. 이 플래그가 0으로 세트되어 있는 경우, 사용자가 PIN 번호를 입력하지 않아

도, 사용자는 그 PlayList를 시청할 수 있다.

<304> write_protect_flag는, 도 28의 A에 테이블을 도시한 바와 같이, 1로 세트되어 있는 경우, write_protect_flag를 제외하고, 그 PlayList의 내용은 소거 및 변경되지 않는다. 이 플래그가 0으로 세트되어 있는 경우, 사용자는 그 PlayList를 자유롭게 소거 및 변경할 수 있다. 이 플래그가 1로 세트되어 있는 경우, 사용자가 그 PlayList를 소거, 편집, 또는 덧씌우기하기 전에, 기록 재생 장치(1)는 사용자에게 재확인하도록 하는 메시지를 표시시킨다.

<305> write_protect_flag가 0으로 세트되어 있는 Real PlayList가 존재하고, 또한, 그 Real PlayList의 Clip를 참조하는 Virtual PlayList가 존재하며, 그 Virtual PlayList의 write_protect_flag가 1로 세트되어 있어도 된다. 사용자가 Real PlayList를 소거하고자 하는 경우, 기록 재생 장치(1)는 그 Real PlayList를 소거하기 전에 상기 Virtual PlayList의 존재를 사용자에게 경고하거나, 또는, 그 Real PlayList를 "최소화"한다.

<306> is_played_flag는, 도 28의 B에 도시한 바와 같이, 플래그가 1로 세트되어 있는 경우, 그 PlayList는 기록되고 나서 한번은 재생된 것을 나타내고, 0으로 세트되어 있는 경우, 그 PlayList는 기록되고 나서 한번도 재생된 적이 없는 것을 나타낸다.

<307> archive는, 도 28의 C에 도시한 바와 같이, 그 PlayList가 오리지널인지, 복사된 것인지를 나타내는 2비트의 필드이다. ref_thumbnail_index의 필드는 PlayList를 대표하는 썸네일 화상의 정보를 나타낸다. ref_thumbnail_index 필드가 0xFFFF가 아닌 값의 경우, 그 PlayList에는 PlayList를 대표하는 썸네일 화상이 부가되어 있고, 그 썸네일 화상은 menu.thum 파일 중에 저장되어 있다. 그 화상은 menu.thum 파일 중에서 ref_thumbnail_index의 값을 이용하여 참조된다. ref_thumbnail_index 필드가 0xFFFF인 경우, 그 PlayList에는 PlayList를 대표하는 썸네일 화상이 부가되어 있지 않다.

<308> 다음으로, PlayItem에 대하여 설명한다. 하나의 PlayItem()은 기본적으로 다음의 데이터를 포함한다. Clip의 파일명을 지정하기 위한 Clip_information_file_name, Clip의 재생 구간을 특정하기 위한 IN_time와 OUT_time의 쌍, PlayList()에 있어서 정의되는 CPI_type이 EP_map type인 경우, IN_time와 OUT_time이 참조하는 부분의 STC_sequence_id 및 선행하는 PlayItem과 현재의 PlayItem과의 접속 상태를 나타내는 부분의 connection_condition이다.

<309> PlayList가 2개 이상의 PlayItem으로 구성될 때, 이들 PlayItem은 PlayList의 글로벌 시간축 상에, 시간의 갭 또는 오버랩없이 일렬로 나란히 배열된다. PlayList()에 있어서 정의되는 CPI_type이 EP_map type이고, 또한, 현재의 PlayItem이 BridgeSequence()를 갖지 않을 때, 그 PlayItem에서 정의되는 IN_time와 OUT_time의 쌍은 STC_sequence_id에 의해 지정되는 동일한 STC 연속 구간 상의 시간을 가리키고 있어야 한다. 이러한 예를 도 29에 도시한다.

<310> 도 30은 PlayList()에 있어서 정의되는 CPI_type이 EP_map type이고, 또한 현재의 PlayItem이 BridgeSequence()를 갖을 때, 다음에 설명하는 규칙이 적용되는 경우를 나타내고 있다. 현재의 PlayItem에 선행하는 PlayItem의 IN_time(도 30에서 IN_time1로 나타내고 있는 것)는 선행하는 PlayItem의 STC_sequence_id에 의해 지정되는 STC 연속 구간 상의 시간을 가리키고 있다. 선행하는 PlayItem의 OUT_time(도 30에서 OUT_time1로 나타내고 있는 것)는 현재의 PlayItem의 BridgeSequenceInfo() 중에서 지정되는 Bridge_Clip 중의 시간을 가리키고 있다. 이 OUT_time은 후술하는 부호화 제한에 따라야 한다.

<311> 현재의 PlayItem의 IN_time(도 30에서 IN_time2로 나타내고 있는 것)는 현재의 PlayItem의 BridgeSequenceInfo() 중에서 지정되는 Bridge_Clip 중의 시간을 가리키고 있다. 이 IN_time도 후술하는 부호화 제한에 따라야 한다. 현재의 PlayItem의 PlayItem의 OUT_time(도 30에서 OUT_time2로 나타내고 있는 것)는 현재의 PlayItem의 STC_sequence_id에 의해 지정되는 STC 연속 구간 상의 시간을 가리키고 있다.

<312> 도 31에 도시한 바와 같이, PlayList()의 CPI_type이 TU_map type인 경우, PlayItem의 IN_time와 OUT_time의 쌍은 동일한 Clip AV 스트림 상의 시간을 가리키고 있다.

<313> PlayItem의 신택스는 도 32에 도시한 바와 같다. 도 32에 도시한 PlayItem의 신택스를 설명하면, Clip_Information_file_name의 필드는 Clip Information file의 파일명을 나타낸다. 이 Clip Information file의 ClipInfo()에 있어서 정의되는 Clip_stream_type은 Clip AV stream을 나타내고 있어야 한다.

<314> STC_sequence_id는 8비트의 필드로, PlayItem이 참조하는 STC 연속 구간의 STC_sequence_id를 나타낸다. PlayList() 중에서 지정되는 CPI_type이 TU_map type인 경우, 이 8비트 필드는 어떠한 의미도 갖지 않고, 0으

로 세트된다. IN_time은 32비트 필드로, PlayItem의 재생 개시 시각을 저장한다. IN_time의 시맨틱스는, 도 33에 도시한 바와 같이, PlayList()에 있어서 정의되는 CPI_type에 따라 다르다.

<315> OUT_time은 32비트 필드로, PlayItem의 재생 종료 시각을 저장한다. OUT_time의 시맨틱스는, 도 34에 도시한 바와 같이, PlayList()에서 정의되는 CPI_type에 따라 다르다.

<316> Connection_Condition은, 도 35에 도시한 바와 같은 선행하는 PlayItem과, 현재의 PlayItem 사이의 접속 상태를 나타내는 2비트의 필드이다. 도 36의 A-D는 도 35에 도시한 Connection_Condition의 각 상태에 대하여 설명하는 도면이다.

<317> 다음으로, BridgeSequenceInfo에 대하여, 도 37을 참조하여 설명한다. BridgeSequenceInfo()는 현재의 PlayItem의 부속 정보로, 다음에 나타내는 정보를 갖는다. Bridge-Clip AV stream 파일과 그것에 대응하는 Clip Information file(도 45)를 지정하는 Bridge_Clip_Information_file_name를 포함한다.

<318> 또한, 선행하는 PlayItem이 참조하는 Clip AV stream 상의 소스 패킷의 어드레스로, 이 소스 패킷에 계속해서 Bridge-Clip AV stream 파일의 최초의 소스 패킷이 접속된다. 이 어드레스는 RSPN_exit_from_previous_Clip라 한다. 또한 현재의 PlayItem이 참조하는 Clip AV stream 상의 소스 패킷의 어드레스로, 이 소스 패킷 전에 Bridge-Clip AV stream 파일의 최후의 소스 패킷이 접속된다. 이 어드레스는 RSPN_enter_to_current_Clip라 한다.

<319> 도 37에서, RSPN_arrival_time_discontinuity는 the Bridge-Clip AV stream 파일 중에서 도착 타임 베이스의 불연속점이 있는 부분의 소스 패킷의 어드레스를 나타낸다. 이 어드레스는 ClipInfo()(도 46) 중에서 정의된다.

<320> 도 38은 BridgeSequenceinfo의 신택스를 나타내는 도면이다. 도 38에 도시한 BridgeSequenceinfo의 신택스를 설명하면, Bridge_Clip_Information_file_name의 필드는, Bridge-Clip AV stream 파일에 대응하는 Clip Information file의 파일명을 나타낸다. 이 Clip Information file의 ClipInfo()에서 정의되는 Clip_stream_type은 'Bridge-Clip AV stream'을 나타내고 있어야 한다.

<321> RSPN_exit_from_previous_Clip의 32비트 필드는 선행하는 PlayItem이 참조하는 Clip AV stream 상의 소스 패킷의 상대 어드레스로, 이 소스 패킷에 계속해서 Bridge-Clip AV stream 파일의 최초의 소스 패킷이 접속된다. RSPN_exit_from_previous_Clip은 소스 패킷 번호를 단위로 하는 크기로, 선행하는 PlayItem이 참조하는 Clip AV stream 파일의 최초의 소스 패킷으로부터 ClipInfo()에 있어서 정의되는 offset_SPN의 값을 초기 값으로 하여 카운트된다.

<322> RSPN_enter_to_current_Clip의 32비트 필드는 현재의 PlayItem이 참조하는 Clip AV stream 상의 소스 패킷의 상대 어드레스로, 이 소스 패킷 전에 Bridge-Clip AV stream 파일의 최후의 소스 패킷이 접속된다. RSPN_exit_from_previous_Clip은 소스 패킷 번호를 단위로 하는 크기로, 현재의 PlayItem이 참조하는 Clip AV stream 파일의 최초의 소스 패킷으로부터 ClipInfo()에서 정의되는 offset_SPN의 값을 초기 값으로 하여 카운트된다.

<323> 다음으로, SubPlayItem에 대하여, 도 39를 참조하여 설명한다. SubPlayItem()의 사용은 PlayList()의 CPI_type이 EP_map type인 경우만 허용된다. 본 실시예에서는, SubPlayItem은 오디오의 후기록의 목적을 위해서만 사용되게 한다. SubPlayItem()은 다음에 나타내는 데이터를 포함한다. 우선, PlayList 중의 sub path가 참조하는 Clip를 지정하기 위한 Clip_information_file_name를 포함한다.

<324> 또한, Clip 중의 sub path의 재생 구간을 지정하기 위한 SubPath_IN_time와 SubPath_OUT_time를 포함한다. 또한, main path의 시간축 상에서 sub path가 재생 개시하는 시간을 지정하기 위한 sync_PlayItem_id와 sync_start_PTS_of_PlayItem을 포함한다. sub path에 참조되는 오디오의 Clip AV stream은 STC 불연속점(시스템 타임 베이스의 불연속점)을 포함해서는 안된다. sub path에 사용되는 Clip의 오디오 샘플의 클럭은 main path의 오디오 샘플의 클럭에 동기되어 있다.

<325> 도 40은 SubPlayItem의 신택스를 나타내는 도면이다. 도 40에 도시한 SubPlayItem의 신택스를 설명하면, Clip_Information_file_name의 필드는 Clip Information file의 파일명을 나타내고, 그것은 PlayList 중에서 sub path에 의해 사용된다. 이 Clip Information file의 ClipInfo()에서 정의되는 Clip_stream_type은 Clip AV stream을 나타내고 있어야 한다.

<326> SubPath_type의 8비트의 필드는 sub path의 타입을 나타낸다. 여기서는, 도 41에 도시한 바와 같이, '0x00'밖

에 설정되어 있지 않고, 다른 값은 장래를 위해 확보되어 있다.

<327> sync_PlayItem_id의 8비트의 필드는 main path의 시간축 상에서 sub path가 재생 개시하는 시각이 포함되는 PlayItem의 PlayItem_id를 나타낸다. 소정의 PlayItem에 대응하는 PlayItem_id의 값은 PlayList()에 있어서 정의된다(도 25 참조).

<328> sync_start_PTS_of_PlayItem의 32비트의 필드는 main path의 시간축 상에서 sub path가 재생 개시하는 시각을 나타내고, sync_PlayItem_id로 참조되는 PlayItem 상의 PTS(Presentation Time Stamp)의 상위 32비트를 나타낸다. SubPath_IN_time의 32비트 필드는 sub path의 재생 개시 시각을 저장한다. SubPath_IN_time은 sub path 중에서 최초의 프레젠테이션 유닛에 대응하는 33비트 길이의 PTS의 상위 32비트를 나타낸다.

<329> SubPath_OUT_time의 32비트 필드는 Sub path의 재생 종료 시각을 저장한다. SubPath_OUT_time은 다음 식에 의해 산출되는 Presentation_end_TS의 값의 상위 32비트를 나타낸다.

<330> Presentation_end_TS=PTS_out+AU_duration

<331> 여기서, PTS_out는 Subpath의 최후의 프레젠테이션 유닛에 대응하는 33비트 길이의 PTS이다. AU_duration은 Subpath의 최후의 프레젠테이션 유닛의 90㎑ 단위의 표시 기간이다.

<332> 다음으로, 도 23에 도시한 xxxxx.rpls와 yyyyy.vpls의 신택스 내의 PlayListMark()에 대하여 설명한다. PlayList에 대한 마크 정보는 이 PlayListMark에 저장된다. 도 42는 PlayListMark의 신택스를 나타내는 도면이다. 도 42에 도시한 PlayListMark의 신택스에 대하여 설명하면, version_number는 이 PlayListMark()의 버전 번호를 나타내는 4개의 캐릭터 문자이다. version_number는 ISO646에 따라 "0045"로 부호화되어야 한다.

<333> length는 이 length 필드의 직후부터 PlayListMark()의 최후까지의 PlayListMark()의 바이트 수를 나타내는 32비트의 부호없는 정수이다. number_of_PlayList_marks는 PlayListMark 중에 저장되어 있는 마크의 개수를 나타내는 16비트의 부호없는 정수이다. number_of_PlayList_marks는 0이라도 무방하다. mark_type은 마크의 타입을 나타내는 8비트의 필드로, 도 43에 도시한 테이블에 따라 부호화된다.

<334> mark_time_stamp의 32비트 필드는 마크가 지정된 포인트를 나타내는 타임 스탬프를 저장한다. mark_time_stamp의 시맨틱스는 도 44에 도시한 바와 같이, PlayList()에서 정의되는 CPI_type에 따라 다르다. PlayItem_id는 마크가 놓여져 있는 부분의 PlayItem을 지정하는 8비트의 필드이다. 소정의 PlayItem에 대응하는 PlayItem_id의 값은 PlayList()에 있어서 정의된다(도 25 참조).

<335> character_set의 8비트의 필드는 mark_name 필드에 부호화되어 있는 캐릭터 문자의 부호화 방법을 나타낸다. 그 부호화 방법은, 도 19에 도시한 값에 대응한다. name_length의 8비트 필드는 mark_name 필드 중에 나타내는 마크명의 바이트 길이를 나타낸다. mark_name의 필드는 마크의 명칭을 나타낸다. 이 필드 중의 좌측으로부터 name_length수의 바이트 수가 유효한 캐릭터 문자로, 그것은 마크의 명칭을 나타낸다. mark_name 필드 중에서, 이들 유효한 캐릭터 문자 뒤의 값은 어떠한 값이 설정되어도 무방하다.

<336> ref_thumbnail_index의 필드는 마크에 부가되는 썸네일 화상의 정보를 나타낸다. ref_thumbnail_index 필드가 0xFFFF가 아닌 값인 경우, 그 마크에는 썸네일 화상이 부가되어 있고, 그 썸네일 화상은 mark.thmb 파일 중에 저장되어 있다. 그 화상은 mark.thmb 파일 중에서 ref_thumbnail_index의 값을 이용하여 참조된다(후술). ref_thumbnail_index 필드가 0xFFFF인 경우, 그 마크에는 썸네일 화상이 부가되어 있지 않은 것을 나타낸다.

<337> 다음으로, Clip information file에 대하여 설명한다. zzzzz.dip(Clip information file 파일)은 도 45에 도시한 바와 같이 6개의 오브젝트로 구성된다. 이들은 ClipInfo(), STC-Info(), ProgramInfo(), CPI(), ClipMark() 및 MakerPrivateData()이다. AV 스트림(Clip AV 스트림 또는 BridgE-Clip AV stream)과 그것에 대응하는 Clip Information 파일은 동일한 숫자 열의 "zzzzz"가 사용된다.

<338> 도 45에 도시한 zzzzz.dip(Clip information file 파일)의 신택스에 대하여 설명하면, ClipInfo_Start_address는 zzzzz.dip 파일의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, ClipInfo()의 선두 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다.

<339> STC_Info_Start_address는 zzzzz.dip 파일의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, STC_Info()의 선두 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다. ProgramInfo_Start_address는 zzzzz.dip 파일의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, ProgramInfo()의 선두 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다. CPI_Start_address는 zzzzz.dip 파일의 선두의 바이트로부터의

상대 바이트 수를 단위로 하여, CPI()의 선두 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다.

<340> ClipMark_Start_address는 zzzzz.dip 파일의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, ClipMark()의 선두 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다. MakerPrivateData_Start_address는 zzzzz.dip 파일의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, MakerPrivateData()의 선두 어드레스를 나타낸다. 상대 바이트 수는 0부터 카운트된다. padding_word(패딩 워드)는 zzzzz.dip 파일의 신택스에 따라 삽입된다. N1, N2, N3, N4 및 N5는 0 또는 임의의 양의 정수이어야 한다. 각각의 패딩 워드는 임의의 값이 취해지도록 해도 된다.

<341> 다음으로, ClipInfo에 대하여 설명한다. 도 46은 ClipInfo의 신택스를 나타내는 도면이다. ClipInfo()는 그것에 대응하는 AV 스트림 파일(Clip AV 스트림 또는 Bridge-Clip AV 스트림 파일)의 속성 정보를 저장한다.

<342> 도 46에 도시한 ClipInfo의 신택스에 대하여 설명하면, version_number는 이 ClipInfo()의 버전 번호를 나타내는 4개의 캐릭터 문자이다. version_number는 ISO646에 따라 "0045"로 부호화되어야 한다. length는 이 length 필드의 직후부터 ClipInfo()의 최후까지의 ClipInfo()의 바이트 수를 나타내는 32비트의 부호없는 정수이다. Clip_stream_type의 8비트의 필드는, 도 47에 도시한 바와 같이, Clip Information 파일에 대응하는 AV 스트림의 타입을 나타낸다. 각각의 타입의 AV 스트림의 스트림 타입에 대해서는 후술한다.

<343> offset_SPN의 32비트의 필드는 AV 스트림(Clip AV 스트림 또는 Bridge-Clip AV 스트림) 파일의 최초의 소스 패킷에 대한 소스 패킷 번호의 오프셋 값을 제공한다. AV 스트림 파일이 최초로 디스크에 기록될 때, 이 offset_SPN은 0이어야 한다.

<344> 도 48에 도시한 바와 같이, AV 스트림 파일의 처음 부분이 편집에 의해 소거되었을 때, offset_SPN은 0 이외의 값을 취해도 된다. 본 실시예에서는, offset_SPN을 참조하는 상대 소스 패킷 번호(상대 어드레스)가 종종 RSPN_xxx(xn은 변형되며 예를 들면 RSPN_EP_start)의 형식으로 신택스 중에 기술되어 있다. 상대 소스 패킷 번호는 소스 패킷 번호를 단위로 하는 크기로, AV 스트림 파일의 최초의 소스 패킷으로부터 offset_SPN의 값을 초기 값으로 하여 카운트된다.

<345> AV 스트림 파일의 최초의 소스 패킷으로부터 상대 소스 패킷 번호로 참조되는 소스 패킷까지의 소스 패킷의 수(SPN_xxx)는 다음 식으로 산출된다.

<346> SPN_xxx=RSPN_xxx-offset_SPN

<347> 도 48은 offset_SPN이 4인 경우의 예를 나타낸다.

<348> TS_recording_rate는 24비트의 부호없는 정수로, 이 값은 DVR 드라이브(기입부(22))로 또는 DVR 드라이브(판독부(28))로부터의 AV 스트림이 필요한 입출력의 비트 레이트를 제공한다. record_time_and_date는 Clip에 대응하는 AV 스트림이 기록되었을 때의 일시를 저장하는 56비트의 필드로, 년/월/일/시/분/초에 대하여, 14개의 숫자를 4비트의 Binary Coded Decimal(BCD)로 부호화한 것이다. 예를 들면, 2001/12/23:01:02:03은 "0x20011223010203"으로 부호화된다.

<349> duration은 Clip의 총 재생 시간을 도착 타임 클럭에 기초한 시간/분/초의 단위로 나타낸 24비트의 필드이다. 이 필드는 6개의 숫자를 4비트의 Binary Coded Decimal(BCD)로 부호화한 것이다. 예를 들면, 01:45:30은 "0x014530"으로 부호화된다.

<350> time_controlled_flag의 플래그는 AV 스트림 파일의 기록 모드를 나타낸다. 이 time_controlled_flag가 1인 경우, 기록 모드는 기록하고 나서의 시간 경과에 대하여 파일 사이즈가 비례하도록 하여 기록되는 모드인 것을 나타내고, 다음 식으로 나타내는 조건을 충족시켜야 한다.

<351> TS_average_rate*192/188*(t-start_time)-α<=size_clip(t)

<352> <=TS_average_rate*192/188*(t-start_time)+α

<353> 여기서, TS_average_rate는 AV 스트림 파일의 전송 스트림의 평균 비트 레이트를 bytes/second의 단위로 나타낸 것이다.

<354> 또한, 상기 식에 있어서, t는 초 단위로 나타내는 시간을 나타내고, start_time은 AV 스트림 파일의 최초의 소스 패킷이 기록되었을 때의 시각으로, 초 단위로 나타낸다. size_clip(t)는 시각 t에서의 AV 스트림 파일의 사이즈를 바이트 단위로 나타낸 것으로, 예를 들면, start_time로부터 시각 t까지 10개의 소스 패킷이 기록된 경

우, size_clip(t)는 10*192바이트이다. α는 TS_average_rate에 의존하는 상수이다.

<355> time_controlled_flag가 0으로 세트되어 있는 경우, 기록 모드는 기록 시간 경과와 AV 스트림의 파일 사이즈가 비례하도록 제어하고 있지 않은 것을 나타낸다. 예를 들면, 이것은 입력 전송 스트림을 트랜스페어런트하게 기록하는 경우이다.

<356> TS_average_rate는 time_controlled_flag가 1로 세트되어 있는 경우, 이 24비트의 필드는 상기 식에서 이용하고 있는 TS_average_rate의 값을 나타낸다. time_controlled_flag가 0으로 세트되어 있는 경우, 이 필드는 어떠한 의미도 갖지 않고 0으로 세트되어야 한다. 예를 들면, 가변 비트 레이트의 전송 스트림은 다음에 나타내는 수순에 의해 부호화된다. 우선 전송 레이트를 TS_recording_rate의 값으로 세트한다. 다음으로, 비디오 스트림을 가변 비트 레이트로 부호화한다. 그리고, 널(nall) 패킷을 사용하지 않는 것에 의해, 간헐적으로 전송 패킷을 부호화한다.

<357> RSPN_arrival_time_discontinuity의 32비트 필드는 Bridge-Clip AV stream 파일 상에서 도착 타임 베이스의 불연속이 발생하는 장소의 상대 어드레스이다. RSPN_arrival_time_discontinuity는 소스 패킷 번호를 단위로 하는 크기로, Bridge-Clip AV stream 파일의 최초의 소스 패킷으로부터 ClipInfo()에 있어서 정의되는 offset_SPN의 값을 초기 값으로 하여 카운트된다. 그 Bridge-Clip AV stream 파일 중에서의 절대 어드레스는 상술한

<358> SPN_xxx=RSPN_xxx-offset_SPN

<359> 에 기초하여 산출된다.

<360> reserved_for_system_use의 144비트의 필드는 시스템용으로 준비되어 있다. is_format_identifier_valid의 플래그가 1일 때, format_identifier의 필드가 유효인 것을 나타낸다. is_original_network_ID_valid의 플래그가 1인 경우, original_network_ID의 필드가 유효인 것을 나타낸다. is_transport_stream_ID_valid의 플래그가 1인 경우, transport_stream_ID의 필드가 유효인 것을 나타낸다. is_service_ID_valid의 플래그가 1인 경우, service_ID의 필드가 유효인 것을 나타낸다.

<361> is_country_code_valid의 플래그가 1일 때, country_code의 필드가 유효인 것을 나타낸다. format_identifier의 32비트 필드는 전송 스트림 중에서 registration deascriotor(ISO/IEC13818-1로 정의되어 있음)가 갖는 format_identifier의 값을 나타낸다. original_network_ID의 16비트 필드는 전송 스트림 중에서 정의되어 있는 original_network_ID의 값을 나타낸다. transport_stream_ID의 16비트 필드는 전송 스트림 중에서 정의되어 있는 transport_stream_ID의 값을 나타낸다.

<362> service_ID의 16비트 필드는, 전송 스트림 중에서 정의되어 있는 service_ID의 값을 나타낸다. country_code의 24비트의 필드는 ISO3166에 의해 정의되는 국가 코드를 나타낸다. 각각의 캐릭터 문자는 ISO8859-1로 부호화된다. 예를 들면, 일본은 "JPN"으로 나타내고, "0x4A 0x50 0x4E"로 부호화된다. stream_format_name는 전송 스트림의 스트림 정의를 하고 있는 포맷 기관의 명칭을 나타내는 ISO-646의 16개의 캐릭터 코드이다. 이 필드 중의 무효한 바이트는 값 '0xFF'가 세트된다.

<363> format_identifier, original_network_ID, transport_stream_ID, service_ID, country_code 및 stream_format_name는 전송 스트림의 서비스 프로바이더를 나타내는 것으로, 이에 따라, 오디오나 비디오 스트림의 부호화 제한, SI(서비스 인포메이션)의 규격이나 오디오 비디오 스트림 이외의 프라이비트 데이터 스트림의 스트림 정의를 인식할 수 있다. 이들 정보는, 디코더가 그 스트림을 디코드할 수 있는지의 여부, 그리고 디코드할 수 있는 경우에 디코드 개시 전에 디코더 시스템의 초기 설정을 행하기 위해 이용하는 것이 가능하다.

<364> 다음으로, STC_Info에 대하여 설명한다. 여기서는, MPEG-2 전송 스트림 중에서 STC의 불연속점(시스템 타임 베이스의 불연속점)을 포함하지 않는 시간 구간을 STC_sequence라 하고, Clip 중에서 STC_sequence는 STC_sequence_id의 값에 의해 특정된다. 도 50의 A 및 B는 연속적인 STC 구간에 대하여 설명하는 도면이다. 동일한 STC_sequence 중에서 동일한 STC의 값은 결코 나타나지 않는다(단, 후술하는 바와 같이, Clip의 최대 시간 길이는 제한되어 있음). 따라서, 동일한 STC_sequence 중에서 동일한 PTS의 값도 또한 결코 나타나지 않는다. AV 스트림이 N(N>0)개의 STC 불연속점을 포함하는 경우, Clip의 시스템 타임 베이스는 (N+1)개의 STC_sequence로 분할된다.

<365> STC_Info는 STC의 불연속(시스템 타임 베이스의 불연속)이 발생하는 장소의 어드레스를 저장한다. 도 51을 참

조하여 설명한 바와 같이, RSPN_STC_start가 그 어드레스를 나타내고, 최후의 STC_sequence를 제외한 k번째(k>=0)의 STC_sequence는 k번째의 RSPN_STC_start로 참조되는 소스 패킷이 도착한 시각으로부터 시작하여, (k+1)번째의 RSPN_STC_start로 참조되는 소스 패킷이 도착한 시각에서 종료된다. 최후의 STC_sequence는 최후의 RSPN_STC_start로 참조되는 소스 패킷이 도착한 시각에서 시작하여, 최후의 소스 패킷이 도착한 시각에서 종료된다.

<366> 도 52는 STC_Info의 신택스를 나타내는 도면이다. 도 52에 도시한 STC_Info의 신택스에 대하여 설명하면, version_number는 이 STC_Info()의 버전 번호를 나타내는 4개의 캐릭터 문자이다. version_number는 ISO646에 따라 "0045"로 부호화되어야 한다.

<367> length는 이 length 필드의 직후부터 STC_Info()의 최후까지의 STC_Info()의 바이트 수를 나타내는 32비트의 부호없는 정수이다. CPI()의 CPI_type이 TU_map type를 나타내는 경우, 이 length 필드는 0을 세트해도 된다. CPI()의 CPI_type이 EP_map type를 나타내는 경우, num_of_STC_sequences는 1 이상의 값이어야 한다.

<368> num_of_STC_sequences의 8비트의 부호없는 정수는 Clip 중에서의 STC_sequence의 수를 나타낸다. 이 값은 이 필드에 계속되는 for-loop의 루프 횟수를 나타낸다. 소정의 STC_sequence에 대응하는 STC_sequence_id는 RSPN_STC_start를 포함하는 for-loop 중에서, 그 STC_sequence에 대응하는 RSPN_STC_start의 나타나는 순서에 의해 정의되는 것이다. STC_sequence_id는 0부터 개시된다.

<369> RSPN_STC_start의 32비트 필드는 AV 스트림 파일 상에서 STC_sequence가 개시하는 어드레스를 나타낸다. RSPN_STC_start는 AV 스트림 파일 중에서 시스템 타임 베이스의 불연속점이 발생하는 어드레스를 나타낸다. RSPN_STC_start는 AV 스트림 중에서 새로운 시스템 타임 베이스의 최초의 PCR을 갖는 소스 패킷의 상대 어드레스로 해도 된다. RSPN_STC_start는 소스 패킷 번호를 단위로 하는 크기로, AV 스트림 파일의 최초의 소스 패킷으로부터 ClipInfo()에 있어서 정의되는 offset_SPN의 값을 초기 값으로 하여 카운트된다. 그 AV stream 파일 중에서의 절대 어드레스는 이미 상술한

<370> SPN_xxx=RSPN_xxx-offset_SPN

<371> 에 의해 산출된다.

<372> 다음으로, 도 45에 도시한 zzzzz.clip의 신택스 내의 ProgramInfo에 대하여 설명한다. 도 53을 참조하면서 설명하면, 여기서는 Clip 중에서 다음의 특징을 갖는 시간 구간을 program_sequence라 한다. 우선, PCR_PID의 값이 변화되지 않는다. 다음으로, 비디오 엘리먼트리 스트림의 수가 변화되지 않는다. 또, 각각의 비디오 스트림에 대한 PID의 값과 그 VideoCodingInfo에 의해 정의되는 부호화 정보가 변화되지 않는다. 또한, 오디오 엘리먼트리 스트림의 수가 변화되지 않는다. 또, 각각의 오디오 스트림에 대한 PID의 값과 그 AudioCodingInfo에 의해 정의되는 부호화 정보가 변화되지 않는다.

<373> program_sequence는 동일한 시각에서 단지 하나의 시스템 타임 베이스를 갖는다. program_sequence는 동일한 시각에서 단지 하나의 PMT를 갖는다. ProgramInfo()는 program_sequence가 개시하는 장소의 어드레스를 저장한다. RSPN_program_sequence_start가 그 어드레스를 나타낸다.

<374> 도 54는 ProgramInfo의 신택스를 나타내는 도면이다. 도 54에 도시한 Program Info의 신택스를 설명하면, version_number는 이 ProgramInfo()의 버전 번호를 나타내는 4개의 캐릭터 문자이다. version_number는 ISO646에 따라 "0045"로 부호화되어야 한다.

<375> length는 이 length 필드의 직후부터 ProgramInfo()의 최후까지의 ProgramInfo()의 바이트 수를 나타내는 32비트의 부호없는 정수이다. CPI()의 CPI_type이 TU_map type를 나타내는 경우, 이 length 필드는 0으로 세트되어도 된다. CPI()의 CPI_type이 EP_map type를 나타내는 경우, number_of_programs는 1 이상의 값이어야 한다.

<376> number_of_program_sequences의 8비트의 부호없는 정수는 Clip 중에서의 Program_sequence의 수를 나타낸다. 이 값은 이 필드에 계속되는 for-loop의 루프 횟수를 나타낸다. Clip 중에서 program_sequence가 변화되지 않은 경우, number_of_program_sequences는 1을 세트해야 한다. RSPN_program_sequence_start의 32비트 필드는 AV 스트림 파일 상에서 프로그램 시퀀스가 개시되는 장소의 상대 어드레스이다.

<377> RSPN_program_sequence_start는 소스 패킷 번호를 단위로 하는 크기로, AV 스트림 파일의 최초의 소스 패킷으로부터 ClipInfo()에서 정의되는 offset_SPN의 값을 초기 값으로 하여 카운트된다. 그 AV 스트림 파일 중에서의

절대 어드레스는

<378> SPN_xxx=RSPN_xxx-offset_SPN

<379> 에 의해 산출된다. 신택스의 for-loop 중에서 RSPN_program_sequence_start 값은 승순으로 나타내어야 한다.

<380> PCR_PID의 16비트 필드는 그 program_sequence에 유효한 PCR 필드를 포함하는 전송 패킷의 PID를 나타낸다. number_of_Videos의 8비트 필드는 video_stream_PID와 VideoCodingInfo()를 포함하는 for-loop의 루프 횟수를 나타낸다. number_of_audios의 8비트 필드는 audio_stream_PID와 AudioCodingInfo()를 포함하는 for-loop의 루프 횟수를 나타낸다. video_stream_PID의 16비트 필드는 그 program_sequence에 유효한 비디오 스트림을 포함하는 전송 패킷의 PID를 나타낸다. 이 필드에 계속되는 VideoCodingInfo()는 그 video_stream_PID로 참조되는 비디오 스트림의 내용을 설명해야 한다.

<381> audio_stream_PID의 16비트 필드는 그 program_sequence에 유효한 오디오 스트림을 포함하는 전송 패킷의 PID를 나타낸다. 이 필드에 계속되는 AudioCodingInfo()는 그 audio_stream_PID로 참조되는 비디오 스트림의 내용을 설명해야 한다.

<382> 또, 신택스의 for-loop 중에서 video_stream_ID의 값이 나타나는 순서는, 그 program_sequence에 유효한 PMT 중에서 비디오 스트림의 PID가 부호화되어 있는 순서와 동일해야 한다. 또한, 신택스의 for-loop 중에서 audio_stream_ID의 값이 나타나는 순서는 그 program_sequence에 유효한 PMT 중에서 오디오 스트림의 PID가 부호화되어 있는 순서와 동일해야 한다.

<383> 도 55는 도 54에 도시한 Programinfo의 신택스 내의 VideoCodingInfo의 신택스를 나타내는 도면이다. 도 55에 도시한 VideoCodingInfo의 신택스를 설명하면, video_format의 8비트 필드는, 도 56에 도시한 바와 같이, ProgramInfo() 중의 video_stream_PID에 대응하는 비디오 포맷을 나타낸다.

<384> frame_rate의 8비트 필드는, 도 57에 도시한 바와 같이, ProgramInfo() 중의 video_stream_PID에 대응하는 비디오의 프레임 레이트를 나타낸다. display_aspect_ratio의 8비트 필드는, 도 58에 도시한 바와 같이, ProgramInfo() 중의 video_stream_PID에 대응하는 비디오의 표시 어스펙트비를 나타낸다.

<385> 도 59는 도 54에 도시한 Programinfo의 신택스 내의 AudioCodingInfo의 신택스를 나타내는 도면이다. 도 59에 도시한 AudioCodingInfo의 신택스를 설명하면, audio_coding의 8비트 필드는, 도 60에 도시한 바와 같이, ProgramInfo() 중의 audio_stream_PID에 대응하는 오디오의 부호화 방법을 나타낸다.

<386> audio_component_type의 8비트 필드는, 도 61에 도시한 바와 같이, ProgramInfo() 중의 audio_stream_PID에 대응하는 오디오의 컴포넌트 타입을 나타낸다. sampling_frequency의 8비트 필드는, 도 62에 도시한 바와 같이, ProgramInfo() 중의 audio_stream_PID에 대응하는 오디오의 샘플링 주파수를 나타낸다.

<387> 다음으로, 도 45에 도시한 zzzzz.clip의 신택스 내의 CPI(Characteristic Point Information)에 대하여 설명한다. CPi는 AV 스트림 중의 시간 정보와 그 파일 중의 어드레스를 관련시키기 위해 있다. CPI에는 2개의 타입이 있으며, 이들은 EP_map과 TU_map이다. 도 63에 도시한 바와 같이, CPI() 중의 CPI_type이 EP_map type인 경우, 그 CPI()는 EP_map을 포함한다. 도 64에 도시한 바와 같이, CPI() 중의 CPI_type이 TU_map type인 경우, 그 CPI()는 TU_map을 포함한다. 하나의 AV 스트림은 하나의 EP_map 또는 하나의 TU_map을 갖는다. AV 스트림이 SESF 전송 스트림인 경우, 그것에 대응하는 Clip은 EP_map을 갖지 않으면 안된다.

<388> 도 65는 CPI의 신택스를 나타내는 도면이다. 도 65에 도시한 CPI의 신택스를 설명하면, version_number는 이 CPI()의 버전 번호를 나타내는 4개의 캐릭터 문자이다. version_number는 ISO646에 따라 "0045"로 부호화되어야 한다. length는 이 length 필드의 직후부터 CPI()의 최후까지의 CPI()의 바이트 수를 나타내는 32비트의 부호없는 정수이다. CPI_type은, 도 66에 도시한 바와 같이, 1비트의 플래그로, Clip의 CPI의 타입을 나타낸다.

<389> 다음으로, 도 65에 도시한 CPI의 신택스 내의 EP_map에 대하여 설명한다. EP_map에는 2개의 타입이 있으며, 그것은 비디오 스트림용의 EP_map과 오디오 스트림용의 EP_map이다. EP_map 중의 EP_map_type이 EP_map의 타입을 구별한다. Clip이 하나 이상의 비디오 스트림을 포함하는 경우, 비디오 스트림용의 EP_map이 사용되어야 한다. Clip이 비디오 스트림을 포함하지 않고, 하나 이상의 오디오 스트림을 포함하는 경우, 오디오 스트림용의 EP_map이 사용되어야 한다.

<390> 비디오 스트림용의 EP_map에 대하여 도 67을 참조하여 설명한다. 비디오 스트림용의 EP_map은 stream_PID, PTS_EP_start 및 RSPN_EP_start라는 데이터를 갖는다. stream_PID는 비디오 스트림을 전송하는 전송 패킷의 PID를 나타낸다. PTS_EP_start는 비디오 스트림의 시퀀스 헤더로부터 시작하는 액세스 유닛의 PTS를 나타낸다. RSPN_EP_start는 AV 스트림 중에서 PTS_EP_start에 의해 참조되는 액세스 유닛의 제1 바이트째를 포함하는 소스 패킷의 어드레스를 나타낸다.

<391> EP_map_for_one_stream_PID()라고 하는 서브 테이블은 동일한 PID를 갖는 전송 패킷에 의해 전송되는 비디오 스트림마다 작성된다. Clip 중에 복수의 비디오 스트림이 존재하는 경우, EP_map은 복수의 EP_map_for_one_stream_PID()를 포함해도 된다.

<392> 오디오 스트림용의 EP_map은 stream_PID, PTS_EP_start 및 RSPN_EP_start라는 데이터를 갖는다. stream_PID는 오디오 스트림을 전송하는 전송 패킷의 PID를 나타낸다. PTS_EP_start는 오디오 스트림의 액세스 유닛의 PTS를 나타낸다. RSPN_EP_start는 AV 스트림 중에서 PTS_EP_start로 참조되는 액세스 유닛의 제1 바이트째를 포함하는 소스 패킷의 어드레스를 나타낸다.

<393> EP_map_for_one_stream_PID()라는 서브 테이블은 동일한 PID를 갖는 전송 패킷에 의해 전송되는 오디오 스트림마다 작성된다. Clip 중에 복수의 오디오 스트림이 존재하는 경우, EP_map은 복수의 EP_map_for_one_stream_PID()를 포함해도 된다.

<394> EP_map와 STC_Info의 관계를 설명하면, 하나의 EP_map_for_one_stream_PID()는 STC의 불연속점에 상관없이 하나의 테이블에 작성된다. RSPN_EP_start의 값과 STC_Info()에 있어서 정의되는 RSPN_STC_start의 값을 비교함으로써, 각각의 STC_sequence에 속하는 EP_map의 데이터의 경계를 알 수 있다(도 68을 참조). EP_map은 동일한 PID로 전송되는 연속된 스트림의 범위에 대하여, 하나의 EP_map_for_one_stream_PID를 갖지 않으면 안된다. 도 69에 도시한 바와 같은 경우, Program#1과 program#3은 동일한 비디오 PID를 갖지만, 데이터 범위가 연속되지 않기 때문에, 각각의 프로그램마다 EP_map_for_one_stream_PID를 갖지 않으면 안된다.

<395> 도 70은 EP_map의 신택스를 나타내는 도면이다. 도 70에 도시한 EP_map의 신택스를 설명하면, EP_type은 4비트의 필드로, 도 71에 도시한 바와 같이, EP_map의 엔트리 포인트 타입을 나타낸다. EP_type은 이 필드에 계속되는 데이터 필드의 시맨틱스를 나타낸다. Clip이 하나 이상의 비디오 스트림을 포함하는 경우, EP_type은 0('video')으로 세트되어야 한다. 또는, Clip이 비디오 스트림을 포함하지 않고, 하나 이상의 오디오 스트림을 포함하는 경우, EP_type은 1('audio')로 세트되어야 한다.

<396> number_of_stream_PIDs의 16비트의 필드는 EP_map() 중의 number_of_stream_PIDs를 변수로 갖는 for-loop의 루프 횟수를 나타낸다. stream_PID(k)의 16비트의 필드는 EP_map_for_one_stream_PID(num_EP_entries(k))에 의해 참조되는 k번째의 엘리먼트리 스트림(비디오 또는 오디오 스트림)을 전송하는 전송 패킷의 PID를 나타낸다. EP_type이 0('video')인 경우, 그 엘리먼트리 스트림은 비디오 스트림이어야 한다. 또한, EP_type이 1('audio')인 경우, 그 엘리먼트리 스트림은 오디오 스트림이어야 한다.

<397> num_EP_entries(k)의 16비트의 필드는 EP_map_for_one_stream_PID(num_EP_entries(k))에 의해 참조되는 num_EP_entries(k)를 나타낸다. EP_map_for_one_stream_PID_Start_address(k)의 32비트의 필드는 EP_map() 중에서 EP_map_for_one_stream_PID(num_EP_entries(k))가 시작되는 상대 바이트 위치를 나타낸다. 이 값은 EP_map()의 제1 바이트째로부터의 크기로 나타낸다.

<398> padding_word는 EP_map()의 신택스에 따라 삽입되어야 한다. X와 Y는 0 또는 임의의 양의 정수이어야 한다. 각각의 패딩 워드는 임의의 값을 취해도 된다.

<399> 도 72는 EP_map_for_one_stream_PID의 신택스를 나타내는 도면이다. 도 72에 도시한 EP_map_for_one_stream_PID의 신택스를 설명하면, PTS_EP_start의 32비트의 필드의 시맨틱스는 EP_map()에서 정의되는 EP_type에 따라 다르다. EP_type이 0('video')인 경우, 이 필드는 비디오 스트림의 시퀀스 헤더에서 시작되는 액세스 유닛의 33비트 정밀도의 PTS의 상위 32비트를 갖는다. EP_type이 1('audio')인 경우, 이 필드는 오디오 스트림의 액세스 유닛의 33비트 정밀도의 PTS의 상위 32비트를 갖는다.

<400> RSPN_EP_start의 32비트의 필드의 시맨틱스는 EP_map()에서 정의되는 EP_type에 따라 다르다. EP_type이 0('video')인 경우, 이 필드는 AV 스트림 중에서 PTS_EP_start에 의해 참조되는 액세스 유닛의 시퀀스 헤더의 제1 바이트째를 포함하는 소스 패킷의 상대 어드레스를 나타낸다. 또는, EP_type이 1('audio')인 경우, 이 필드는 AV 스트림 중에서 PTS_EP_start에 의해 참조되는 액세스 유닛의 오디오 프레임의 제1 바이트째를 포함하는

소스 패킷의 상대 어드레스를 나타낸다.

<401> RSPN_EP_start는 소스 패킷 번호를 단위로 하는 크기로, AV 스트림 파일의 최초의 소스 패킷으로부터 ClipInfo()에서 정의되는 offset_SPN의 값을 초기 값으로 하여 카운트된다. 그 AV 스트림 파일 중에서의 절대 어드레스는

<402> SPN_xxx=RSPN_xxx-offset_SPN

<403> 에 의해 산출된다. 신택스의 for-loop 중에서 RSPN_EP_start의 값은 승순으로 나타내야 한다.

<404> 다음으로, TU_map에 대하여, 도 73을 참조하여 설명한다. TU_map은 소스 패킷의 도착 타임 클럭(도착 시각 베이스의 시계)에 기초하여, 하나의 시간축을 작성한다. 그 시간축은 TU_map_time_axis라 한다. TU_map_time_axis의 원점은 TU_map() 중의 offset_time에 의해 나타낸다. TU_map_time_axis는 offset_time로부터 일정한 단위로 분할된다. 그 단위를, time_unit이라 한다.

<405> AV 스트림 중의 각각의 time_unit 중에서 최초의 완전한 형태의 소스 패킷의 AV 스트림 파일 상의 어드레스가 TU_map에 저장된다. 이들 어드레스를 RSPN_time_unit_start라 한다. TU_map_time_axis 상에서, k(k>=0)번째의 time_unit가 시작되는 시각은 TU_start_time(k)라 한다. 이 값은 다음 식에 기초하여 산출된다.

<406> TU_start_time(k)=offset_time+k*time_unit_size

<407> TU_start_time(k)는 45㎒의 정밀도를 갖는다.

<408> 도 74는 TU_map의 신택스를 나타내는 도면이다. 도 74에 도시한 TU_map의 신택스를 설명하면, offset_time의 32비트 길이의 필드는 TU_map_time_axis에 대한 오프셋 타임을 제공한다. 이 값은 Clip 중의 최초의 time_unit에 대한 오프셋 시각을 나타낸다. offset_time은 27㎒ 정밀도의 도착 타임 클럭으로부터 도출되는 45㎑ 클럭을 단위로 하는 크기이다. AV 스트림이 새로운 Clip으로서 기록되는 경우, offset_time은 0으로 세트되어야 한다.

<409> time_unit_size의 32비트 필드는 time_unit의 크기를 제공하는 것으로, 그것은 27㎒ 정밀도의 도착 타임 클럭으로부터 도출되는 45㎒ 클럭을 단위로 하는 크기이다. time_unit_size는 1초 이하(time_unit_size<=45000)로 하는 것이 좋다. number_of_time_unit_entries의 32비트 필드는 TU_map() 중에 저장되어 있는 time_unit의 엔트리 수를 나타낸다.

<410> RSPN_time_unit_start의 32비트 필드는 AV 스트림 중에서 각각의 time_unit가 개시되는 장소의 상대 어드레스를 나타낸다. RSPN_time_unit_start는 소스 패킷 번호를 단위로 하는 크기로, AV stream 파일의 최초의 소스 패킷으로부터 ClipInfo()에 있어서 정의되는 offset_SPN의 값을 초기 값으로 하여 카운트된다. 그 AV stream 파일 중에서의 절대 어드레스는

<411> SPN_xxx=RSPN_xxx-offset_SPN

<412> 에 의해 산출된다. 신택스의 for-loop 중에서 RSPN_time_unit_start의 값은 승순으로 나타내야 한다. (k+1)번째의 time_unit 중에 소스 패킷이 아무것도 없는 경우, (k+1)번째의 RSPN_time_unit_start는 k번째의 RSPN_time_unit_start와 같아야 한다.

<413> 도 45에 도시한 zzzzz.clip의 신택스 내의 ClipMark에 대하여 설명한다. ClipMark는 클립에 대한 마크 정보로, ClipMark 중에 저장된다. 이 마크는 기록기(기록 재생 장치(1))에 의해 세트되는 것으로, 사용자에 의해 세트되는 것이 아니다.

<414> 도 75는 ClipMark의 신택스를 나타내는 도면이다. 도 75에 도시한 ClipMark의 신택스를 설명하면, version_number는 이 ClipMark()의 버전 번호를 나타내는 4개의 캐릭터 문자이다. version_number는 ISO646에 따라 "0045"로 부호화되어야 한다.

<415> length는 이 length 필드의 직후부터 ClipMark()의 최후까지의 ClipMark()의 바이트 수를 나타내는 32비트의 부호없는 정수이다. number_of_Clip_marks는 0이라도 무방하다. mark_type은 마크의 타입을 나타내는 8비트의 필드로, 도 76에 도시한 테이블에 따라 부호화된다.

<416> mark_time_stamp는 32비트 필드로, 마크가 지정된 포인트를 나타내는 타임 스탬프를 저장한다. mark_time_stamp의 시맨틱스는, 도 77에 도시한 바와 같이, PlayList() 중의 CPI_type에 따라 다르다.

<417> STC_sequence_id는 CPI() 중의 CPI_type이 EP_map type를 나타내는 경우, 이 8비트의 필드는 마크가 놓여져 있

는 부분의 STC 연속 구간의 STC_sequence_id를 나타낸다. CPI() 중의 CPI_type이 TU_map type를 나타내는 경우, 이 8비트의 필드는 어떠한 의미도 갖지 않고, 0으로 세트된다. character_set의 8비트의 필드는 mark_name 필드에 부호화되어 있는 캐릭터 문자의 부호화 방법을 나타낸다. 그 부호화 방법은 도 19에 나타내는 값에 대응한다.

<418> name_length의 8비트 필드는 Mark_name 필드 중에 나타내는 마크명의 바이트 길이를 나타낸다. mark_name의 필드는 마크의 명칭을 나타낸다. 이 필드 중의 좌측으로부터 name_length수의 바이트 수가 유효한 캐릭터 문자로, 그것은 마크의 명칭을 나타낸다. mark_name 필드 중에서, 이들 유효한 캐릭터 문자 뒤의 값은 어떤 값이 들어가도 무방하다.

<419> ref_thumbnail_index의 필드는 마크에 부가되는 썸네일 화상의 정보를 나타낸다. ref_thumbnail_index 필드가 0xFFFF가 아닌 값인 경우, 그 마크에는 썸네일 화상이 부가되어 있고, 그 썸네일 화상은 mark.thmb 파일 중에 저장되어 있다. 그 화상은 mark.thmb 파일 중에서 ref_thumbnail_index의 값을 이용하여 참조된다. ref_thumbnail_index 필드가 0xFFFF인 경우, 그 마크에는 썸네일 화상이 부가되어 있지 않다.

<420> 도 78은 도 75 대신에 ClipMark의 다른 신택스를 나타내는 도면이고, 도 79는 그 경우에서의 도 76 대신에 mark_type의 테이블의 예를 나타낸다. reserved_for_maker_ID는 mark_type이 0xC0으로부터 0xFF의 값을 나타낼 때, 그 mark_type를 정의하고 있는 메이커의 메이커 ID를 나타내는 16비트의 필드이다. 메이커 ID는 DVR 포맷 라이센서가 지정한다. mark_entry()는 마크점에 지정된 포인트를 나타내는 정보이고, 그 신택스의 상세는 후술한다. representative_picture_entry()는 mark_entry()에 의해 나타내는 마크를 대표하는 화상의 포인트를 나타내는 정보로, 그 신택스의 상세는 후술한다.

<421> ClipMark는 사용자가 AV 스트림을 재생할 때, 그 내용을 시각적으로 검색할 수 있도록 하기 위해 이용된다. DVR 플레이어는 GUI(그래피컬 사용자 인터페이스)를 사용하여, ClipMark의 정보를 사용자에게 제시한다. ClipMark의 정보를 시각적으로 표시하기 위해서는, mark_entry()가 나타내는 픽쳐보다 오히려 representative_picture_entry()가 나타내는 픽쳐를 나타낸 쪽이 좋다.

<422> 도 80은 mark_entry()와 representative_picture_entry()의 예를 나타낸다. 예를 들면, 임의의 프로그램이 개시되고 나서, 잠시 후(수초 후)에, 그 프로그램의 프로그램명(타이틀)이 표시되도록 한다. ClipMark를 작성할 때는, mark_entry()는 그 프로그램의 개시 포인트에 놓고, representative_picture_entry()는 그 프로그램의 프로그램명(타이틀)이 표시되는 포인트에 놓도록 해도 된다.

<423> DVR 플레이어는 representative_picture_entry의 화상을 GUI에 표시하고, 사용자가 그 화상을 지정하면, DVR 플레이어는 mark_entry가 놓인 포인트로부터 재생을 개시한다.

<424> 도 81은 mark_entry() 및 representative_picture_entry()의 신택스를 나타낸다.

<425> mark_time_stamp는 32비트 필드로, mark_entry()의 경우에는 마크가 지정된 포인트를 나타내는 타임 스탬프를 저장하고, 또한 representative_picture_entry()의 경우, mark_entry()에 의해 나타내는 마크를 대표하는 화상의 포인트를 나타내는 타임 스탬프를 저장한다.

<426> 다음으로, ClipMark를 지정하기 위해, PTS에 의한 타임 스탬프 베이스의 정보를 사용하는 것이 아니라, 어드레스 베이스의 정보를 사용하는 경우의 mark_entry()와 representative_picture_entry()의 신택스의 예를 도 82에 도시한다.

<427> RSPN_ref_EP_start는, mark_entry()의 경우, AV 스트림 중에서 마크점의 픽쳐를 디코드하기 위한 스트림의 엔트리 포인트를 나타내는 소스 패킷의 상대 어드레스를 나타낸다. 또한, representative_picture_entry()의 경우, mark_entry()에 의해 나타내는 마크를 대표하는 픽쳐를 디코드하기 위한 스트림의 엔트리 포인트를 나타내는 소스 패킷의 상대 어드레스를 나타낸다. RSPN_ref_EP_start의 값은 EP_map 중에 RSPN_EP_start로서 저장되어 있어야 하고, 또한, 그 RSPN_EP_start에 대응하는 PTS_EP_start의 값은 EP_map 중에서 마크점의 픽쳐의 PTS보다 과거에 가장 가까운 값이어야 한다.

<428> offset_num_pictures는 32비트의 필드로, RSPN_ref_EP_start에 의해 참조되는 픽쳐로부터 표시 순서대로 마크점으로 나타내는 픽쳐까지의 오프셋의 픽쳐 수를 나타낸다. 이 수는 0부터 카운트된다. 도 83의 예의 경우, offset_num_pictures는 6으로 된다.

<429> 다음으로, ClipMark를 지정하기 위해, 어드레스 베이스의 정보를 사용하는 경우의 mark_entry()와

representative_picture_entry()의 신택스의 다른 예를 도 84에 도시한다.

<430> RSPN_mark_point는, mark_entry()의 경우, AV 스트림 중에서 그 마크가 참조하는 액세스 유닛의 제1 바이트째를 포함하는 소스 패킷의 상대 어드레스를 나타낸다. 또한, representative_picture_entry()의 경우, mark_entry()에 의해 나타내는 마크를 대표하는 부호화 픽쳐의 제1 바이트째를 포함하는 소스 패킷의 상대 어드레스를 나타낸다.

<431> RSPN_mark_point는 소스 패킷 번호를 단위로 하는 크기로, AV 스트림 파일의 최초의 소스 패킷으로부터 Clip Information file에서 정의되는 offset_SPN의 값을 초기 값으로 하여 카운트된다.

<432> 도 85를 참조하여, ClipMark와 EP_map의 관계를 설명한다. 이 예의 경우, EP_map이 엔트리 포인트의 어드레스로서 I0, I1, In을 지정하고 있고, 이들 어드레스로부터 시퀀스 헤더로 이어지는 I픽쳐가 개시되도록 한다. ClipMark가 임의의 마크의 어드레스로서 M1을 지정하고 있을 때, 그 소스 패킷으로부터 개시하고 있는 픽쳐를 디코드할 수 있기 위해서는, M1의 어드레스보다 앞에서 가장 가까운 엔트리 포인트인 I1로부터 데이터를 판독 개시하면 된다.

<433> MakersPrivateData에 대해서는, 도 22를 참조하여 이미 설명하였기 때문에, 그 설명은 생략한다.

<434> 다음으로, 썸네일 인포메이션(Thumbnail Information)에 대하여 설명한다. 썸네일 화상은 menu.thmb 파일 또는 mark.thmb 파일로 저장된다. 이들 파일은 동일한 신택스 구조로, 단 하나의 Thumbnail()을 갖는다. menu.thmb 파일은 메뉴 썸네일 화상 즉 Volume을 대표하는 화상 및 각각의 PlayList를 대표하는 화상을 저장한다. 모든 메뉴 썸네일은 단 하나의 menu.thmb 파일로 저장된다.

<435> mark.thmb 파일은 마크 썸네일 화상, 즉 마크점을 나타내는 픽쳐를 저장한다. 모든 PlayList 및 Clip에 대한 모든 마크 썸네일은 단지 하나의 mark.thmb 파일로 저장된다. 썸네일은 빈번히 추가, 삭제되기 때문에, 추가 조작과 부분 삭제 조작은 용이하게 고속으로 실행할 수 있어야 한다. 이 이유 때문에, Thumbnail()은 블록 구조를 갖는다. 화상의 데이터는 몇 개의 부분으로 분할되고, 각 부분은 하나의 tn_block으로 저장된다. 하나의 화상 데이터는 연속된 tn_block으로 저장된다. tn_block의 열에는 사용되고 있지 않은 tn_block가 존재해도 된다. 하나의 썸네일 화상의 바이트 길이는 가변이다.

<436> 도 86은 menu.thmb와 mark.thmb의 신택스를 나타내는 도면이고, 도 87은 도 86에 도시한 menu.thmb와 mark.thmb의 신택스 내의 Thumbnail의 신택스를 나타내는 도면이다. 도 87에 도시한 Thumbnail의 신택스에 대하여 설명하면, version_ number는 이 Thumbnail()의 버전 번호를 나타내는 4개의 캐릭터 문자이다. version_number는 ISO646에 따라 "0045"로 부호화되어야 한다.

<437> length는 이 length 필드의 직후부터 Thumbnail()의 최후까지의 MakersP rivateData()의 바이트 수를 나타내는 32비트의 부호없는 정수이다. tn_blocks_start_address는 Thumbnail()의 선두의 바이트로부터의 상대 바이트 수를 단위로 하여, 최초의 tn_block의 선두 바이트 어드레스를 나타내는 32비트의 부호없는 정수이다. 상대 바이트 수는 0부터 카운트된다. number_of_thumbnails는 Thumbnail() 중에 포함되어 있는 썸네일 화상의 엔트리 수를 제공하는 16비트의 부호없는 정수이다.

<438> tn_block_size는 1024바이트를 단위로 하여, 하나의 tn_block의 크기를 제공하는 16비트의 부호없는 정수이다. 예를 들면, tn_block_size=1이면, 그것은 하나의 tn_block의 크기가 1024바이트인 것을 나타낸다. number_of_tn_blocks는 이 Thumbnail() 중의 tn_block의 엔트리 수를 나타내는 116비트의 부호없는 정수이다. thumbnail_index는 이 thumbnail_index 필드로부터 시작되는 for_loop 1회분의 썸네일 정보로 나타내는 썸네일 화상의 인덱스 번호를 나타내는 16비트의 부호없는 정수이다. thumbnail_index로서, 0xFFFF라는 값을 사용해서는 안된다. thumbnail_index는 UIAppInfoVolume(), UIAppInfoPlayList(), PlayListMark() 및 ClipMark() 중의 ref-thumbnail_index에 의해 참조된다.

<439> thumbnail_picture_format는 썸네일 화상의 픽쳐 포맷을 나타내는 8비트의 부호없는 정수로, 도 80에 도시한 바와 같은 값을 취한다. 표 중의 DCF와 PNG는 "menu.thmb" 내에서만 허용된다. 마크 썸네일은 값 "0x00"(MPEG-2 Video I-picture)을 취해야만 한다.

<440> picture_data_size는 썸네일 화상의 바이트 길이를 바이트 단위로 나타내는 32비트의 부호없는 정수이다. start_tn_block_number은 썸네일 화상의 데이터가 시작되는 tn_block의 tn_block 번호를 나타내는 16비트의 부호없는 정수이다. 썸네일 화상 데이터의 선두는 tn_block의 선두와 일치해야 한다. tn_block 번호는 0부터 시작하여, tn_block의 for-loop 중의 변수 k의 값에 관계된다.

<441> x_picture_length는 썸네일 화상의 프레임 화면의 수평 방향의 픽셀 수를 나타내는 16비트의 부호없는 정수이다. y_picture_length는 썸네일 화상의 프레임 화면의 수직 방향의 픽셀 수를 나타내는 16비트의 부호없는 정수이다. tn_block는 썸네일 화상이 저장되는 영역이다. Thumbnail() 중의 모든 tn_block는 동일한 사이즈(고정 길이)로, 그 크기는 tn_block_size에 의해 정의된다.

<442> 도 89A 및 도 89B는 썸네일 화상 데이터가 어떻게 tn_block에 저장되는지를 모식적으로 나타낸 도면이다. 도 89A 및 도 89B와 같이, 각 썸네일 화상 데이터는 tn_block의 선두로부터 시작하여, tn_block를 넘는 크기인 경우에는 연속하는 다음의 tn_block를 사용하여 저장된다. 이와 같이 함으로써, 가변 길이인 픽쳐 데이터가 고정 길이의 데이터로서 관리하는 것이 가능해지고, 삭제 편집에 대하여 간편한 처리로 대응할 수 있게 된다.

<443> 다음으로, AV 스트림 파일에 대하여 설명한다. AV 스트림 파일은 "M2TS" 디렉토리(도 14)에 저장된다. AV 스트림 파일에는 2개의 타입이 있으며, 그들은 Clip AV 스트림과 Bridge-Clip AV 스트림 파일이다. 이들 AV 스트림은 모두 이 이후에 정의되는 DVR MPEG-2 전송 스트림 파일의 구조이어야 한다.

<444> 우선, DVR MPEG-2 전송 스트림에 대하여 설명한다. DVR MPEG-2 전송 스트림의 구조는, 도 90에 도시한 바와 같이 되어 있다. AV 스트림 파일은 DVR MPEG-2 전송 스트림의 구조를 갖는다. DVR MPEG-2 전송 스트림은 정수개의 Aligned unit로 구성된다. Aligned unit의 크기는 6144바이트(2048*3바이트)이다. Aligned unit는 소스 패킷의 제1 바이트째로부터 시작된다. 소스 패킷은 192바이트 길이이다. 하나의 소스 패킷은 TP_extra_header와 전송 패킷으로 이루어진다. TP_extra_header는 4바이트 길이이고, 또한 전송 패킷은 188바이트 길이이다.

<445> 하나의 Aligned unit는 32개의 소스 패킷으로 이루어진다. DVR MPEG-2 전송 스트림 중의 최후의 Aligned unit도 또한 32개의 소스 패킷으로 이루어진다. 따라서, DVR MPEG-2 전송 스트림은 Aligned unit의 경계에서 종단한다. 디스크에 기록되는 입력 전송 스트림의 전송 패킷의 수가 32의 배수가 아닐 때, 널 패킷(PID=0x1FFF의 전송 패킷)을 갖는 소스 패킷을 최후의 Aligned unit에 사용해야 한다. 파일 시스템은 DVR MPEG-2 전송 스트림에 여분의 정보를 부가해서는 안된다.

<446> 도 91은 DVR MPEG-2 전송 스트림의 레코더 모델을 나타낸다. 도 91에 도시한 레코더는 레코딩 프로세스를 규정하기 위한 개념상의 모델이다. DVR MPEG-2 전송 스트림은 이 모델에 따른다.

<447> MPEG-2 전송 스트림의 입력 타이밍에 대해 설명한다. 입력 MPEG-2 전송 스트림은 전체 전송 스트림 또는 부분 전송 스트림이다. 입력되는 MPEG-2 전송 스트림은 ISO/IEC13818-1 또는 ISO/IEC13818-9에 따라야 한다. MPEG-2 전송 스트림의 i번째의 바이트는 T-STD(ISO/IEC 13818-1로 규정되는 Transport stream system target decoder)(51)와 소스 패킷타이저(sourse packetizer)(54)로 시각 t(i)에서 동시에 입력된다. Rpk는 전송 패킷의 입력 레이트의 순간적인 최대치이다.

<448> 27㎒ PLL(52)은 27㎒ 클럭의 주파수를 발생한다. 27㎒ 클럭의 주파수는 MPEG-2 전송 스트림의 PCR(Program Clock Reference)의 값에 동기된다. 도착 타임 클럭 카운터(53)는 27㎒의 주파수의 펄스를 카운트하는 2진 카운터이다. Arrival_time_clock(i)는 시각 t(i)에서의 도착 타임 클럭 카운터의 카운트 값이다.

<449> 소스 패킷타이저(54)는 모든 전송 패킷에 TP_extra_header를 부가하여, 소스 패킷을 작성한다. Arrival_time_stamp는 전송 패킷의 제1 바이트째가 T-STD와 소스 패킷타이저의 양방에 도착하는 시각을 나타낸다. Arrival_time_stamp(k)는 다음 식으로 나타낸 바와 같이 Arrival_time_clock(k)의 샘플치이고, 여기서, k는 전송 패킷의 제1 바이트째를 나타낸다.

<450> arrival_time_stamp(k)=arrival_time_clock(k)%230

<451> 2개의 연속하여 입력되는 전송 패킷의 시간 간격이 230/27000000초(약40초) 이상으로 되는 경우, 그 2개의 전송 패킷의 arrival_time_stamp의 차분은 230/27000000초가 되도록 세트되어야 한다. 레코더는 그와 같이 되는 경우에 구비되어 있다.

<452> 스무징 버퍼(55)는 입력 전송 스트림의 비트 레이트를 스무징한다. 스무징 버퍼는 오버플로우해서는 안된다. Rmax는 스무징 버퍼가 비어 있지 않을 때의 스무징 버퍼로부터의 소스 패킷의 출력 비트 레이트이다. 스무징 버퍼가 비어 있을 때, 스무징 버퍼로부터의 출력 비트 레이트는 0이다.

<453> 다음으로, DVR MPEG-2 전송 스트림의 레코더 모델의 파라미터에 대하여 설명한다. Rmax라는 값은 AV 스트림 파일에 대응하는 ClipInfo()에서 정의되는 TS_recording_rate에 의해 부여된다. 이 값은, 다음 식에 의해 산출된

다.

<454> Rmax=TS_recording_rate*192/188

<455> TS_recording_rate의 값은 바이트/초를 단위로 하는 크기이다.

<456> 입력 전송 스트림이 SESF 전송 스트림인 경우, Rpk는 AV 스트림 파일에 대응하는 ClipInfo()에서 정의되는 TS_recording_rate와 같아야 한다. 입력 전송 스트림이 SESF 전송 스트림이 아닌 경우, 이 값은 MPEG-2 전송 스트림의 기술자, 예를 들면 maximum_bitrate_descriptor나 partial_transport_stream_descriptor 등에서 정의되는 값을 참조하여도 된다.

<457> 입력 트랜스 스트림이 SESF 전송 스트림인 경우, 스무징 버퍼(55)의 크기는 0이다. 입력 전송 스트림이 SESF 전송 스트림이 아닌 경우, 스무징 버퍼의 크기는 MPEG-2 전송 스트림의 기술자, 예를 들면 smoothing_buffer_descriptor, short_smoothing_buffer_descriptor, partial_transport_stream_descriptor 등에서 정의되는 값을 참조하여도 된다.

<458> 기록기(레코더) 및 재생기(플레이어)는 충분한 사이즈의 버퍼를 준비해야 한다. 디폴트 버퍼의 기본 사이즈는 1536바이트이다.

<459> 다음으로, DVR MPEG-2 전송 스트림의 플레이어 모델에 대하여 설명한다. 도 92는 DVR MPEG-2 전송 스트림의 플레이어 모델을 나타내는 도면이다. 이것은 재생 프로세스를 규정하기 위한 개념 상의 모델이다. DVR MPEG-2 전송 스트림은 이 모델에 따른다.

<460> 27㎒ X-tal61은 27㎒의 주파수를 발생한다. 27㎒ 주파수의 오차 범위는, +/-30ppm(27000000+/-810Hz)이 아니면 안된다. 도착 타임 클럭 카운터(62)는 27㎒의 주파수의 펄스를 카운트하는 2진 카운터이다. Arrival_time_clock(i)은 시각 t(i)에서의 도착 타임 클럭 카운터의 카운트 값이다.

<461> 스무징 버퍼(64)에서, Rmax는 스무징 버퍼가 충만 상태가 아닐 때의 스무징 버퍼에의 소스 패킷의 입력 비트 레이트이다. 스무징 버퍼가 충만 상태일 때 스무징 버퍼로의 입력 비트 레이트는 0이다.

<462> MPEG-2 전송 스트림의 출력 타이밍을 설명하면, 현재의 소스 패킷의 arrival_time_stamp가 arrival_time_clock(i)의 LSB 30비트의 값과 같을 때, 그 소스 패킷의 전송 패킷은 스무징 버퍼로부터 방출된다. Rpk는 전송 패킷 레이트의 순간적인 최대치이다. 스무징 버퍼는 언더플로우해서는 안된다.

<463> DVR MPEG-2 전송 스트림의 플레이어 모델의 파라미터에 대해서는, 상술한 DVR MPEG-2 전송 스트림의 레코더 모델의 파라미터와 동일하다.

<464> 도 93은 소스 패킷(source packet)의 신택스를 나타내는 도면이다. transport_packet()은 ISO/IEC 13818-1로 규정되는 MPEG-2 전송 패킷이다. 도 93에 도시한 소스 패킷의 신택스 내의 TP_Extra_header의 신택스를 도 94에 도시한다. 도 94에 도시한 TP_Extra_header의 신택스에 대하여 설명하면, copy_permission_indicator는 전송 패킷의 페이로드의 복사 제한을 나타내는 정수이다. 복사 제한은 copy free, no more copy, copy once, 또는 copy prohibited로 할 수 있다. 도 95는 copy_permission_indicator의 값과, 이들에 의해 지정되는 모드의 관계를 나타낸다.

<465> copy_permission_indicator는 모든 전송 패킷에 부가된다. IEEE1394 디지털 인터페이스를 사용하여 입력 전송 스트림을 기록하는 경우, copy_permission_indicator의 값은 IEEE1394 아이소크로너스 패킷 헤더(isochronous packet header) 중의 EMI(Encryption Mode Indicator)의 값에 관련시켜도 된다. IEEE1394 디지털 인터페이스를 사용하지 않고 입력 전송 스트림을 기록하는 경우, copy_permission_indicator의 값은 전송 패킷 중에 매립된 CCI의 값에 관련시켜도 된다. 아날로그 신호 입력을 셀프 인코드하는 경우, copy_permission_indicator의 값은 아날로그 신호의 CGMS_A의 값에 관련시켜도 무방하다.

<466> arrival_time_stamp는, 다음 식

<467> arrival_time_stamp(k)=arrival_time_clock(k)% 230

<468> 에서, arrival_time_stamp에 의해 지정되는 값을 갖는 정수값이다.

<469> Clip AV 스트림의 정의를 하는데 있어서, Clip AV 스트림은 상술한 바와 같은 정의가 되는 DVR MPEG-2 전송 스트림의 구조를 갖지 않으면 안된다. arrival_time_clock(i)은, Clip AV 스트림의 중에서 연속하여 증가해야만 된다. Clip AV 스트림 중에 시스템 타임 베이스(STC 베이스)의 불연속점이 존재하였다고 해도, 그 Clip AV 스

트림의 arrival_time_ clock(i)은 연속하여 증가해야만 된다.

<470> Clip AV 스트림의 중의 개시와 종료 사이의 arrival_time_clock(i)의 차분의 최대치는 26시간이 아니면 안된다. 이 제한은, MPEG2 전송 스트림 중에 시스템 타임 베이스(STC 베이스)의 불연속점이 존재하지 않는 경우에, Clip AV 스트림 중에서 동일한 값의 PTS(Presentation Time Stamp)가 절대로 나타나지 않는다는 사실을 보증한다. MPEG2 시스템 규격은, PTS의 랩어라운드(wraparound) 주기를 233/90000초(약 26.5 시간)로 규정하고 있다.

<471> Bridge-Clip AV 스트림의 정의를 하기 위해, Bridge-Clip AV 스트림은, 상술한 바와 같은 정의가 되는 DVR MPEG-2 전송 스트림의 구조를 갖지 않으면 안된다. Bridge-Clip AV 스트림은, 하나의 도착 타임 베이스의 불연속점을 포함해야 한다. 도착 타임 베이스의 불연속점의 전후의 전송 스트림은 후술하는 부호화의 제한에 따라야 하고, 또한 후술하는 DVR-STD 에 따라야 한다.

<472> 본 실시예에서는, 편집에서의 PlayItem 사이의 비디오와 오디오의 심리스 접속을 지원한다. PlayItem 사이를 심리스 접속으로 하는 것은, 플레이어/레코더에 "데이터의 연속 공급"과 "심리스 복호 처리"를 보증한다. "데이터의 연속 공급"이란, 파일 시스템이 디코더에 버퍼의 언더플로우를 초래하지 않도록 필요한 비트 레이트로 데이터를 공급하는 것을 보증할 수 있는 것이다. 데이터의 실시간성을 보증하여, 데이터를 디스크로부터 판독할 수 있도록 데이터가 충분한 크기의 연속된 블록 단위로 저장되도록 한다.

<473> "심리스 복호 처리"란, 플레이어가 디코더의 재생 출력에 일시 중지(pause)나 끊김(gap)을 유발시키지 않고, 디스크에 기록된 오디오 비디오 데이터를 표시할 수 있는 것이다.

<474> 심리스 접속되어 있는 PlayItem이 참조하는 AV 스트림에 대하여 설명한다. 선행하는 PlayItem과 현재의 PlayItem의 접속이, 심리스 표시할 수 있도록 보증되어 있는지의 여부는 현재의 PlayItem에서 정의되어 있는 connection_condition 필드로부터 판단할 수 있다. PlayItem 사이의 심리스 접속은 Bridge-Clip을 사용하는 방법과 사용하지 않는 방법이 있다.

<475> 도 96은 Bridge-Clip을 사용하는 경우의 선행하는 PlayItem과 현재의 PlayItem의 관계를 나타내고 있다. 도 96에서는, 플레이어가 판독하는 스트림 데이터를, 그림자를 넣어 나타내고 있다. 도 96에 도시한 TS1은 Clip1(Clip AV 스트림)의 그림자를 넣은 스트림 데이터와 Bridge-Clip의 RSPN_arrival_time_discontinuity 보다 이전의 그림자를 넣은 스트림 데이터로 이루어진다.

<476> TS1의 Clip1의 그림자를 넣은 스트림 데이터는, 선행하는 PlayItem의 IN_time(도 96에서 IN_time1로 도시되어 있음)에 대응하는 프레젠테이션 유닛을 복호하기 위해 필요한 스트림의 어드레스로부터, RSPN_exit_from_previous_Clip으로 참조되는 소스 패킷까지의 스트림 데이터이다. TS1에 포함되는 Bridge-Clip의 RSPN_arrival_time_discontinuity 보다 이전의 그림자를 넣은 스트림 데이터는, Bridge-Clip의 최초의 소스 패킷으로부터 RSPN_arrival_time_discontinuity로 참조되는 소스 패킷의 직전의 소스 패킷까지의 스트림 데이터이다.

<477> 또한, 도 96에서의 TS2는, Clip2(Clip AV 스트림)의 그림자를 넣은 스트림 데이터와 Bridge-Clip의 RSPN_arrival_time_discontinuity 이후의 그림자를 넣은 스트림 데이터로 이루어진다. TS2에 포함되는 Bridge-Clip의 RSPN_arrival_time_discontinuity 이후의 그림자를 넣은 스트림 데이터는, RSPN_arrival_time_discontinuity로 참조되는 소스 패킷으로부터, Bridge-Clip의 최후의 소스 패킷까지의 스트림 데이터이다. TS2의 Clip2의 그림자를 넣은 스트림 데이터는, RSPN_enter_to_current_Clip으로 참조되는 소스 패킷으로부터, 현재의 PlayItem의 OUT_time(도 96에서 OUT_time2로 도시되어 있음)에 대응하는 프레젠테이션 유닛을 복호하기 위해 필요한 스트림의 어드레스까지의 스트림 데이터이다.

<478> 도 97은 Bridge-Clip을 사용하지 않은 경우의 선행하는 PlayItem과 현재의 PlayItem의 관계를 나타내고 있다. 이 경우, 플레이어가 판독하는 스트림 데이터는, 그림자를 넣어 나타내고 있다. 도 97에서의 TS1은, Clip1(Clip AV 스트림)의 그림자를 붙인 스트림 데이터로 이루어진다. TS1의 Clip1의 그림자를 넣은 스트림 데이터는 선행하는 PlayItem의 IN_time(도 97에서 IN_time1로 도시되어 있음)에 대응하는 프레젠테이션 유닛을 복호하기 위해 필요한 스트림의 어드레스로부터 시작되어 Clip1의 최후의 소스 패킷까지의 데이터이다. 또한, 도 97에서의 TS2는, Clip2(Clip AV 스트림)의 그림자를 붙인 스트림 데이터로 이루어진다.

<479> TS2의 Clip2의 그림자를 넣은 스트림 데이터는, Clip2의 최초의 소스 패킷으로부터 시작하여, 현재의 PlayItem의 OUT_time(도 97에서 OUT_ime2로 도시되어 있음)에 대응하는 프레젠테이션 유닛을 복호하기 위해 필요한 스트림의 어드레스까지의 스트림 데이터이다.

<480> 도 96과 도 97에서, TS1과 T2는 소스 패킷의 연속한 스트림이다. 다음으로, TS1과 TS2의 스트림 규정과, 이들 사이의 접속 조건에 대하여 고려한다. 우선, 심리스 접속을 위한 부호화 제한에 대하여 고려한다. 전송 스트림의 부호화 구조의 제한으로서, 우선, TS1과 TS2 중에 포함되는 프로그램의 수는 1이어야 한다. TS1과 TS2 중에 포함되는 비디오 스트림의 수는 1이어야 한다. TS1과 TS2 중에 포함되는 오디오 스트림의 수는 2 이하이어야 한다. TS1과 TS2 중에 포함되는 오디오 스트림의 수는 같아야 한다. TS1 및/또는 TS2 중에, 상기 이외의 엘리먼트리 스트림 또는 프라이비트 스트림이 포함되어 있어도 무방하다.

<481> 비디오 비트 스트림의 제한에 대하여 설명한다. 도 98은 픽쳐의 표시 순서에서 나타내는 심리스 접속의 예를 나타내는 도면이다. 접속점에서 비디오 스트림을 심리스하게 표시할 수 있기 위해서는, OUT_time1(Clip1의 OUT_time) 뒤와 IN_time2(Clip2의 IN_time)의 앞에 표시되는 불필요한 픽쳐는 접속점 부근의 Clip의 부분적인 스트림을 재차 인코드하는 프로세스에 의해 제거되어야 한다.

<482> 도 98에 도시한 바와 같은 경우에서, BridgeSequence를 사용하여 심리스 접속을 실현하는 예를, 도 99에 도시한다. RSPN_arrival_time_discontinuity 보다 앞의 Bridge-Clip의 비디오 스트림은, 도 98의 Clip1의 OUT_time1에 대응하는 픽쳐까지의 부호화 비디오 스트림으로 이루어진다. 그리고, 그 비디오 스트림은 선행하는 Clip1의 비디오 스트림에 접속되고, 하나의 연속으로 MPEG2 규격에 따른 엘리먼트리 스트림이 되도록 재차 인코드되어 있다.

<483> 마찬가지로 하여, RSPN_arrival_time_discontinuity 이후의 Bridge-Clip의 비디오 스트림은, 도 98의 Clip2의 IN_time2에 대응하는 픽쳐 이후의 부호화 비디오 스트림으로 이루어진다. 그리고, 그 비디오 스트림은 정확하게 디코드 개시할 수 있어, 이것에 후속되는 Clip2의 비디오 스트림에 접속되고, 하나의 연속으로 MPEG2 규격에 따른 엘리먼트리 스트림이 되도록 재차 인코드되어 있다. Bridge-Clip를 만들기 위해서는, 일반적으로 여러장의 픽쳐는 재차 인코드해야 하고, 그 이외의 픽쳐는 오리지널의 Clip으로부터 복사할 수 있다.

<484> 도 98에 도시한 예의 경우에 BridgeSequence를 사용하지 않고 심리스 접속을 실현하는 예를 도 100에 도시한다. Clip1의 비디오 스트림은, 도 98의 OUT_time1에 대응하는 픽쳐까지의 부호화 비디오 스트림으로 이루어지고, 그것은 하나의 연속으로 MPEG2 규격에 따른 엘리먼트리 스트림이 되도록 재차 인코드되어 있다. 마찬가지로 하여, Clip2의 비디오 스트림은 도 98의 Clip2의 IN-time2에 대응하는 픽쳐 이후의 부호화 비디오 스트림으로 이루어지고, 그것은 하나의 연속으로 MPEG2 규격에 따른 엘리먼트리 스트림이 되도록 재차 인코드되어 있다.

<485> 비디오 스트림의 부호화 제한에 대하여 설명하면, 우선, TS1과 TS2의 비디오 스트림의 프레임 레이트는 같아야 한다. TS1의 비디오 스트림은 sequence_end_code로 종단해야 한다. TS2의 비디오 스트림은, Sequence Header, GOP Header, 그리고 I-픽쳐로 개시해야 한다. TS2의 비디오 스트림은, 클로즈드 GOP로 개시해야 한다.

<486> 비트 스트림 중에서 정의되는 비디오 프레젠테이션 유닛(프레임 또는 필드)은 접속점을 두어 연속이어야 한다. 접속점에서, 프레임 또는 필드의 갭이 있어서는 안된다. 접속점에서, 상한의 필드 시퀀스는 연속이여야만 한다. 3-2 풀다운을 사용하는 인코드의 경우에는, "top_field_first" 및 "repeat_first_field" 플래그를 재기입할 필요가 있을지도 모르며, 또는 필드 갭의 발생을 방지하기 위해 국소적으로 재차 인코드하도록 하여도 좋다.

<487> 오디오 비트 스트림의 부호화 제한에 대하여 설명하면, TS1과 TS2의 오디오의 샘플링 주파수는 동일해야만 한다. TS1과 TS2의 오디오의 부호화 방법(예: MPEG1 레이어2, AC-3, SESF LPCM, AAC)은 동일해야만 한다.

<488> 다음으로, MPEG-2 전송 스트림의 부호화 제한에 대하여 설명하면, TS1의 오디오 스트림의 최후의 오디오 프레임은, TS1의 최후의 표시 픽쳐의 표시 종료 시각과 같은 표시 시각을 갖는 오디오 샘플을 포함하고 있어야 한다. TS2의 오디오 스트림의 최초의 오디오 프레임은 TS2의 최초의 표시 픽쳐의 표시 개시 시각과 동일한 표시 시각을 갖는 오디오 샘플을 포함하고 있어야 한다.

<489> 접속점에서, 오디오 프레젠테이션 유닛의 시퀀스에 갭이 있어서는 안된다. 도 101에 도시한 바와 같이, 2 오디오 프레임 구간 미만의 오디오 프레젠테이션 유닛의 길이로 정의되는 오버랩이 있어도 된다. TS2의 엘리먼트리 스트림을 전송하는 최초의 패킷은 비디오 패킷이어야 한다. 접속점에서의 전송 스트림은 후술하는 DVR-STD에 따라야 한다.

<490> Clip 및 Bridge-Clip의 제한에 대하여 설명하면, TS1과 TS2는 각각에 라이벌 타임 베이스의 불연속점을 포함해야 한다.

<491> 이하의 제한은, Bridge-Clip을 사용하는 경우에만 적용된다. TS1의 최후의 소스 패킷과 TS2의 최초의 소스 패킷의 접속점에서만, Bridge-Clip AV 스트림은 단 하나의 도착 타임 베이스의 불연속점을 갖는다. ClipInfo()에서 정의되는 RSPN_arrival_time_discontinuity가 그 불연속점의 어드레스를 나타내며, 그것은 TS2의 최초의 소스 패킷을 참조하는 어드레스를 나타내야 한다.

<492> BridgeSequenceInfo()에서 정의되는 RSPN_exit_from_previous_Clip에 의해 참조되는 소스 패킷은 Clip1 중의 어떤 소스 패킷이어도 된다. 그것은, Aligned unit의 경계일 필요는 없다. BridgeSequenceInfo()에서 정의되는 RSPN_enter_to_current_Clip에 의해 참조되는 소스 패킷은, Clip2 중의 어떤 소스 패킷이어도 된다. 그것은, Aligned unit의 경계일 필요는 없다.

<493> PlayItem의 제한에 대하여 설명하면, 선행하는 PlayItem의 OUT_time(도 96,도 97에서 도시되는 OUT_time1)은, TS1의 최후의 비디오 프레젠테이션 유닛의 표시 종료 시각을 나타내야 한다. 현재의 PlayItem의 IN_time(도 96, 도 97에서 도시되는 IN_time2)은 TS2의 최초의 비디오 프레젠테이션 유닛의 표시 개시 시각을 나타내야 한다.

<494> Bridge_Clip을 사용하는 경우의 데이터 할당의 제한에 대하여, 도 102를 참조하여 설명하면, 심리스 접속은 파일 시스템에 의해 데이터의 연속 공급이 보증되도록 만들어야 한다. 이것은 Clip1(Clip AV 스트림 파일)과 Clip2(Clip AV 스트림 파일)에 접속되는 Bridge-Clip AV 스트림을 데이터 할당 규정을 충족시키도록 배치함으로써 행해야만 한다.

<495> RSPN_exit_from_previous_Clip 이전의 Clip1(Clip AV 스트림 파일)의 스트림 부분이 하프 프래그먼트(half fragment) 이상의 연속 영역에 배치되어 있도록, RSPN_exit_from_previous_Clip이 선택되어야 한다. Bridge_Clip AV 스트림의 데이터 길이는 하프 프래그먼트 이상의 연속 영역에 배치되도록, 선택되어야 한다. RSPN_enter_to_current_Clip 이후의 Clip2(Clip AV 스트림 파일)의 스트림 부분이 하프 프래그먼트 이상의 연속 영역에 배치되어 있도록, RSPN_enter_to_current-Clip이 선택되어야 한다.

<496> Bridge-Clip을 사용하지 않고 심리스 접속하는 경우의 데이터 할당의 제한에 대하여, 도 103을 참조하여 설명하면, 심리스 접속은 파일 시스템에 의해 데이터의 연속 공급이 보증되도록 만들어야 한다. 이것은, Clip1(Clip AV 스트림 파일)의 최후의 부분과 Clip2(Clip AV 스트림 파일)의 최초의 부분을, 데이터 할당 규정을 충족시키도록 배치함으로써 행해야 한다.

<497> Clip1(Clip AV 스트림 파일)의 최후의 스트림 부분이 하프 프래그먼트 이상의 연속 영역에 배치되어야 한다. Clip2(Clip AV 스트림 파일)의 최초의 스트림 부분이, 하프 프래그먼트 이상의 연속 영역에 배치되어야 한다.

<498> 다음으로, DVR-STD에 대하여 설명한다. DVR-STD는 DVR MPEG2 전송 스트림의 생성 및 검증 시에서의 디코드 처리를 모델화하기 위한 개념 모델이다. 또한, DVR-STD는 상술한 심리스 접속된 2개의 PlayItem에 의해 참조되는 AV 스트림의 생성 및 검증 시에서의 디코드 처리를 모델화하기 위한 개념 모델이기도 하다.

<499> DVR-STD 모델을 도 104에 도시한다. 도 104에 도시한 모델에는, DVR MPEG-2 전송 스트림 플레이어 모델이 구성 요소로서 포함되어 있다. n, TBn, MBn, EBn, TBsys, Bsys, Rxn, Rbxn, Rxsys, Dn, Dsys, On 및 Pn(k)의 표기 방법은 ISO/IEC13818-1의 T-STD에 정의되어 있는 것과 동일하다. 즉, 다음에 설명하는 바와 같다. n은 엘리먼트리 스트림의 인덱스 번호이다. TBn은 엘리먼트리 스트림n의 전송 버퍼이다.

<500> MBn은 엘리먼트리 스트림 n의 다중 버퍼이다. 비디오 스트림에 대해서만 존재한다. EBn은 엘리먼트리 스트림 n의 엘리먼트리 스트림 버퍼이다. 비디오 스트림에 대해서만 존재한다. TBsys는 복호 중의 프로그램의 시스템 정보를 위한 입력 버퍼이다. Bsys는 복호 중의 프로그램의 시스템 정보를 위한 시스템 타깃 디코더 내의 메인 버퍼이다. Rxn은 데이터가 TBn으로부터 제거되는 전송 레이트이다. Rbxn은 PES 패킷 페이로드가 MBn으로부터 제거되는 전송 레이트이다. 비디오 스트림에 대해서만 존재한다.

<501> Rxsys는 데이터가 TBsys로부터 제거되는 전송 레이트이다. Dn은 엘리먼트리 스트림 n의 디코더이다. Dsys는 복호 중의 프로그램의 시스템 정보에 관한 디코더이다. On은 비디오 스트림 n의 re_ordering buffer이다. Pn(k)는 엘리먼트리 스트림 n의 k번째의 프레젠테이션 유닛이다.

<502> DVR-STD의 디코딩 프로세스에 대하여 설명한다. 단일의 DVR MPEG-2 전송 스트림을 재생하고 있는 동안에는, 전송 패킷을 TB1, TBn 또는 TBsys의 버퍼로 입력하는 타이밍은 소스 패킷의 arrival_time_stamp에 의해 결정된다. TB1, MB1, EB1, TBn, Bn, TBsys 및 Bsys의 버퍼링 동작의 규정은, ISO/IEC 13818-1에 규정되어 있는 T-STD와

동일하다. 복호 동작과 표시 동작의 규정도 또한, ISO/IEC 13818-1에 규정되어 있는 T-STD와 동일하다.

<503> 심리스 접속된 PlayItem을 재생하고 있는 동안의 디코딩 프로세스에 대하여 설명한다. 여기서는, 심리스 접속된 PlayItem에 의해 참조되는 2개의 AV 스트림의 재생에 대하여 설명을 하도록 하고, 이후의 설명에서는, 상술한(예를 들면, 도 96에 도시한) TS1과 TS2의 재생에 대하여 설명한다. TS1은 선행하는 스트림이고, TS2는 현재의 스트림이다.

<504> 도 105는, 임의의 AV 스트림(TS1)으로부터 그것에 심리스하게 접속된 다음의 AV 스트림(TS2)으로 이행할 때의 전송 패킷의 입력, 복호, 표시의 타이밍차트를 나타낸다. 소정의 AV 스트림(TS1)으로부터 그것에 심리스하게 접속된 다음의 AV 스트림(TS2)으로 이행하는 동안에는, TS2의 도착 타임 베이스의 시간축(도 105에서 ATC2로 나타냄)은 TS1의 도착 타임 베이스의 시간축(도 105에서 ATC1로 나타냄)과 동일하지 않다.

<505> 또한, TS2의 시스템 타임 베이스의 시간축(도 105에서 STC2로 나타냄)은, TS1의 시스템 타임 베이스의 시간축(도 105에서 STC1로 나타냄)과 동일하지는 않다. 비디오의 표시는 심리스하게 연속되어 있는 것이 요구된다. 오디오의 프레젠테이션 유닛의 표시 시간에는 오버랩이 있어도 무방하다.

<506> DVR-STD로의 입력 타이밍에 대하여 설명한다. 시각 T1까지의 시간, 즉, TS1의 최후의 비디오 패킷이 DVR-STD의 TB1에 입력 종료할 때까지는 DVR-STD의 TB1, TBn 또는 TBsys의 버퍼로의 입력 타이밍은 TS1의 소스 패킷의 arrival_time_stamp에 의해 결정된다.

<507> TS1의 남은 패킷은 TS_recording_rate(TS1)의 비트 레이트로 DVR-STD의 TBn또는 TBsys의 버퍼로 입력되어야 한다. 여기서, T8_recording_rate(TS1)는 Clip1에 대응하는 ClipInfo()에서 정의되는 TS_recording_rate의 값이다. TS1의 최후의 바이트가 버퍼로 입력되는 시각은 시각 T2이다. 따라서, 시각 T1 내지 T2까지의 구간에서는, 소스 패킷의 arrival_time_stamp는 무시된다.

<508> N1을 TS1의 최후의 비디오 패킷에 계속되는 TS1의 전송 패킷의 바이트 수로 하면, 시각 T1 내지 T2까지의 시간 DT1은, N1 바이트가 TS_recording_rate(TS1)의 비트 레이트로 입력 종료하기 위해 필요한 시간이며, 다음 식에 의해 산출된다.

<509> $\Delta T1=T2-T1=N1/TS\_recording\_rate$

<510> (TS1) 시각 T1 내지 T2까지의 사이는 RXn과 RXsys의 값은 모두, TS_recording_rate(TS1)의 값으로 변화한다. 이 규칙 이외의 버퍼링 동작은 T-STD 와 동일하다.

<511> T2의 시각에서, 도착 타임 클럭 카운터는 TS2의 최초의 소스 패킷의 arrival_time_stamp의 값으로 리세트된다. DVR-STD의 TB1, TBn 또는 TBsys의 버퍼에의 입력 타이밍은 TS2의 소스 패킷의 arrival_time_stamp에 의해 결정된다. RXn과 RXsys는 모두 T-STD에서 정의되어 있는 값으로 변화한다.

<512> 부가적인 오디오 버퍼링 및 시스템 데이터 버퍼링에 대하여 설명하면, 오디오 디코더와 시스템 디코더는 시각 T1 내지 T2까지의 구간의 입력 데이터를 처리할 수 있도록, T-STD로 정의되는 버퍼량 외에 부가적인 버퍼량(약 1초분의 데이터량)이 필요하다.

<513> 비디오의 프레젠테이션 타이밍에 대하여 설명하면, 비디오 프레젠테이션 유닛의 표시는 접속점을 통해 갭없이 연속이어야 한다. 여기서, STC1은 TS1의 시스템 타임 베이스의 시간축(도 105에서는 STC1로 도시되어 있음)으로 하고, STC2는 TS2의 시스템 타임 베이스의 시간축(도 105에서는 STC2로 도시되어 있음. 정확하게는, STC2는 TS2의 최초의 PCR이 T-STD에 입력한 시각에서 개시함)으로 한다.

<514> STC1과 STC2의 사이의 오프셋은 다음과 같이 결정된다. PTS1end는 TS1의 최후의 비디오 프레젠테이션 유닛에 대응하는 STC1 상의 PTS이고, PTS2start는, TS2의 최초의 비디오 프레젠테이션 유닛에 대응하는 STC2 상의 PTS이고, Tpp는 TS1의 최후의 비디오 프레젠테이션 유닛의 표시 기간으로 하면, 2개의 시스템 타임 베이스의 사이의 오프셋 STC-delta는 다음 식에 의해 산출된다.

<515> $STC\_delta=PTS1end+Tpp-PT82start$

<516> 오디오의 프레젠테이션의 타이밍에 대하여 설명하면, 접속점에서, 오디오 프레젠테이션 유닛의 표시 타이밍의 오버랩이 있어도 되고, 그것은 0 내지 2 오디오 프레임 미만이다 (도 105에 도시되어 있는 "audio overlap"을 참조). 어느 한쪽의 오디오 샘플을 선택하는가 하는 점과, 오디오 프레젠테이션 유닛의 표시를 접속점의 후의 보정된 타임 베이스에 재동기하는 것은, 플레이어측에 의해 설정되는 것이다.

<517> DVR-STD의 시스템 타임 클럭에 대하여 설명하면, 시각 T5에서, TS1의 최후의 오디오 프렌젠테이션 유닛이 표시된다. 시스템 타임 클럭은, 시각 T2 내지 T5의 사이에 오버랩하고 있어도 된다. 이 구간에서는, DVR-STD는 시스템 타임 클럭을 오래된 타임 베이스의 값(STC1)과 새로운 타임 베이스의 값(STC2) 사이에서 전환한다. STC2의 값은 다음 식에 의해 산출된다.

<518> STC2=STC1-STC_delta

<519> 버퍼링의 연속성에 대하여 설명한다. STC11video_end는, TS1의 최후의 비디오 패킷의 최후의 바이트가 DVR-STD의 TB1로 도착할 때의 시스템 타임 베이스 STC1상의 STC의 값이다. STC22video_start는, TS2의 최초의 비디오 패킷의 최초의 바이트가 DVR-STD의 TB1로 도착할 때의 시스템 타임 베이스 STC2 상의 STC의 값이다. STC21video_end는 STC11video_end의 값을 시스템 타임 베이스 STC2 상의 값으로 환산한 값이다. STC21video_end는 다음 식에 의해 산출된다.

<520> STC21video_end=STC11video_end-STC_delta

<521> DVR-STD에 따르기 위해, 다음의 2개의 조건을 충족시키는 것이 요구된다. 우선, TS2의 최초의 비디오 패킷의 TB1로의 도착 타이밍은, 다음에 나타내는 부등식을 충족하여야 한다. 그리고, 다음에 나타내는 부등식을 충족하여야 한다.

<522> STC22video_start>STC21video_end+ΔT1

<523> 상기 부등식이 충족되도록, Clip1 및, 또는, Clip2의 부분적인 스트림을 재차 인코드 및, 또는, 재차 다중화할 필요가 있는 경우에는, 그 필요에 따라 행해진다.

<524> 다음으로, STC1과 STC2를 동일한 시간축 상에 환산한 시스템 타임 베이스의 시간축 상에 있어서, TS1로부터의 비디오 패킷의 입력과 그것에 계속되는 TS2로부터의 비디오 패킷의 입력은, 비디오 버퍼를 오버플로우 및 언더플로우시켜서는 안된다.

<525> 이러한 신택스, 데이터 구조, 규칙에 기초함으로써, 기록 매체에 기록되어 있는 데이터의 내용, 재생 정보 등을 적절하게 관리할 수가 있기 때문에, 사용자가 재생 시에 적절하게 기록 매체에 기록되어 있는 데이터의 내용을 확인하거나, 원하는 데이터를 간편하게 재생할 수 있도록 할 수 있다.

<526> 또, 본 실시예는, 다중화 스트림으로서 MPEG-2 전송 스트림을 예로 하여 설명하고 있지만, 이것에 한하지 않고, MPEG2 프로그램 스트림이나 미국의 DirecTV 서비스(상표)로 사용되어 있는 DSS 전송 스트림에 대해서도 적용하는 것이 가능하다.

<527> 다음으로, mark_entry() 및 representative_picture_entry()의 신택스가, 도 81에 나타내는 바와 같은 구성인 경우에 있어서의 마크점으로 나타내는 장면의 재생 개시를 행하는 경우의 처리에 대하여, 도 106의 흐름도를 참조하여 설명한다.

<528> 처음에 단계 S1에서, 기록 재생 장치(1)의 제어부(23)는 기록 매체(100)로부터 DVR 전송 스트림 파일의 데이터베이스인 EP_map(도 70), STC_Info(도 52), Program_Info(도 54) 및 ClipMark(도 78)를 판독한다.

<529> 단계 S2에서, 제어부(23)는 ClipMark(도 78)의 representative_picture_entry(도 81) 또는 ref-thumbnail_index로 참조되는 픽쳐로부터 썸네일의 리스트를 작성하고, 사용자 인터페이스 입출력으로서의 단자(24)로부터 출력하여, GUI의 메뉴 화면상에 표시시킨다. 이 경우, ref-thumbnail_index가 유효한 값을 갖는 경우, representative_picture_entry보다 ref_thumbnail_index가 우선된다.

<530> 단계 S3에서, 사용자가 재생 개시점의 마크점을 지정한다. 이것은, 예를 들면, GUI로서 표시된 메뉴 화면상 중에서 사용자가 썸네일 화상을 선택함으로써 행해진다. 제어부(23)는 이 선택 조작에 대응하여, 지정된 썸네일에 대응되어 있는 마크점을 취득한다.

<531> 단계 S4에서, 제어부(23)는 단계 S3에서 지정된 mark_entry(도 81)의 mark_time_stamp의 PTS와, STC_sequence_id를 취득한다.

<532> 단계 S5에서, 제어부(23)는 STC_Info(도 52)로부터 단계 S4에서 취득한 STC_sequence_id에 대응하는 STC 시간축이 개시하는 소스 패킷 번호를 취득한다.

<533> 단계 S6에서, 제어부(23)는 단계 S5에서 취득한 STC 시간축이 개시하는 패킷 번호와, 단계 S4에서 취득한 마크점의 PTS로부터, 마크점의 PTS보다 시간적으로 앞이며, 또한, 가장 가까운 엔트리 포인트(I픽쳐)가 있는 소스

패킷 번호를 취득한다.

<534> 단계 S7에서, 제어부(23)는 단계 S6에서 취득한 엔트리 포인트가 있는 소스 패킷 번호로부터 전송 스트림의 데이터를 판독하고, AV 디코더(27)에 공급시킨다.

<535> 단계 S8에서, 제어부(23)는 AV 디코더(27)를 제어하여, 단계 S4에서 취득한 마크점의 PTS의 픽쳐로부터 표시를 개시시킨다.

<536> 이상의 동작을 도 107 내지 109를 참조하여 다시 설명한다.

<537> 지금, 도 107에 도시한 바와 같이, DVR 전송 스트림 파일은 STC_sequence_id=id0의 STC 시간축을 갖고, 그 시간축이 개시하는 소스 패킷 번호는 장면 개시점 A의 소스 패킷 번호보다 작은 것으로 한다. 그리고, 소스 패킷 번호 B에서 C까지의 사이에, CM(커머셜)이 삽입되어 있는 것으로 한다.

<538> 이 때, 도 70에 나타내는 EP_map에 대응하는 EP_map에는, 도 108에 도시한 바와 같이, RSPN_EP_start로 나타내는 A, B, C에 대응하여, 각각의 PTS가 PTS_EP_start로서 PTS(A), PTS(B), PTS(C)로서 등록된다.

<539> 또한, 도 109에 도시된 바와 같이, 도 78의 ClipMark에 대응하는 ClipMark에는, 도 109에 도시된 바와 같이, 장면 개시, CM 개시 및 CM 종료를 나타내는 마크 타입(도 79) 0x92, 0x94, 0x95의 값에 대응하여, mark_entry와 representative_picture_entry가 기록된다.

<540> mark_ entry의 Mark_Time_stamp로서는, 장면 개시, CM 개시 및 CM 종료에 대응하여, 각각 PTS(a1), PTS(b0), PTS(c0)이 등록되어 있고, 각각의 STC_sequence_id는 모두 id0으로 되어 있다.

<541> 마찬가지로, representative_picture_entry의 Mark_Time_stamp로서, 장면 개시, CM 개시 및 CM 종료에 대응하여, 각각 PTS(a2), PTS(b0), PTS(c0)가 등록되어 있고, 그들은 모두 STC_sequence_id가 id0으로 되어 있다.

<542> PTS(A)<PTS(a1)의 경우, 단계 S6에서, 패킷 번호 A가 취득되고, 단계 S7에서, 패킷 번호 A로부터 시작되는 전송 스트림이 AV 디코더(27)에 공급되며, 단계 S8에서, PTS(a1)의 픽쳐로부터 표시가 개시된다.

<543> 다음으로, 도 110의 흐름도를 참조하여, mark_entry와 representative_picture_entry의 신택스가 도 81에 도시한 바와 같은 구성인 경우에 있어서의 CM 스킵 재생의 처리에 대하여, 도 110의 흐름도를 참조하여 설명한다.

<544> 단계 S21에서, 제어부(23)는 EP_map(도 70), STC_Info(도 52), Program_Info(도 54) 및 ClipMark(도 78)를 기록 매체(100)로부터 판독한다. 단계 S22에서, 사용자는 사용자 인터페이스 입출력으로서의 단자(24)로부터 CM 스킵 재생을 지정한다.

<545> 단계 S23에서, 제어부(23)는 마크 타입(도 79)이 CM 개시점(0x94)인 마크 정보의 PTS와, CM 종료점(0x95)인 마크 정보의 PTS 및 대응하는 STC_sequence_id를 취득한다(도 81).

<546> 단계 S24에서, 제어부(23)는 STC_Info(도 52)로부터 CN 개시점과 종료점의 STC_sequence_id에 대응하는 STC 시간축이 개시하는 소스 패킷 번호를 취득한다.

<547> 단계 S25에서, 제어부(23)는 기록 매체(100)로부터 전송 스트림을 판독하고, 그것을 AV 디코더(27)에 공급하여 디코드를 개시시킨다.

<548> 단계 S26에서, 제어부(23)는 현재의 표시 화상이 CM 개시점의 PTS의 화상인지의 여부를 조사한다. 현재의 표시 화상이 CM 개시점의 PTS의 화상이 아닌 경우에는, 단계 S27로 진행하여, 제어부(23)는 화상의 표시를 계속한다. 그 후, 처리는 단계 S25로 되돌아가, 그 이후의 처리가 반복되어 실행된다.

<549> 단계 S26에서, 현재의 표시 화상이 CM 개시점의 PTS의 화상이다라고 판정된 경우, 단계 S28로 진행하여, 제어부(23)는 AV 디코더(27)를 제어하여, 디코드 및 표시를 정지시킨다.

<550> 다음으로, 단계 S29에서, 제어부(23)는 CM 종료점의 STC_sequence_id에 대응하는 STC 시간축이 개시하는 패킷 번호를 취득하고, 그 패킷 번호와, 단계 S23의 처리에서 취득한 CM 종료점의 PTS로부터 그 점의 PTS보다 시간적으로 앞이며, 또한, 가장 가까운 엔트리 포인트가 있는 소스 패킷 번호를 취득한다.

<551> 단계 S30에서, 제어부(23)는 단계 S29의 처리에서 취득한 엔트리 포인트가 있는 소스 패킷 번호로부터 전송 스트림의 데이터를 판독하고, AV 디코더(27)에 공급시킨다.

<552> 단계 S31에서, 제어부(23)는 AV 디코더(27)를 제어하고, CM 종료점의 PTS의 픽쳐로부터 표시를 재개시킨다.

<553> 도 107 내지 도 109를 참조하여, 이상의 동작을 더 설명하면, CM 개시점과 CM 종료점은, 이 예의 경우, STC_sequence_id=id0이라는 공통의 STC 시간축 상에 존재하고, 그 STC 시간축이 개시하는 소스 패킷 번호는 장면의 개시점의 소스 패킷 번호 A보다 작은 것으로 되어 있다.

<554> 전송 스트림이 디코드되고, 단계 S26에서 표시 시각이 PTS(b0)로 되었다고 판정된 경우(CM 개시점이다라고 판정된 경우), AV 디코더(27)에 의해 표시가 정지된다. 그리고, PTS(C)<PTS(c0)의 경우, 단계 S30에서 패킷 번호 C의 데이터로부터 시작되는 스트림으로부터 디코드가 재개되고, 단계 S31에서, PTS(c0)의 픽쳐로부터 표시가 재개된다.

<555> 또, 이 방법은 CM 스킵 재생에 한하지 않고, 일반적으로 ClipMark로 지정되는 2점 사이의 장면을 스킵하여 재생하는 경우에도 적용 가능하다.

<556> 다음으로, mark_entry와 representative_picture_entry가, 도 82에 나타내는 신택스 구조인 경우에 있어서의 마크점으로 나타내는 CM의 검색 재생 처리에 대하여, 도 112의 흐름도를 참조하여 설명한다.

<557> 단계 S41에서, 제어부(23)는 EP_map(도 70), STC_Info(도 52), Program_Info(도 54) 및 ClipMark(도 78)의 정보를 취득한다.

<558> 다음으로 단계 S42에서, 제어부(23)는 단계 S41에서 판독한 ClipMark(도 78)에 포함되는 representative_picture_entry(도 82) 또는 ref_thumbnail_index로 참조되는 픽쳐로부터 썸네일의 리스트를 생성하고, GUI의 메뉴 화면상에 표시시킨다. ref_thumbnail_index가 유효한 값을 갖는 경우, representative_picture_entry보다 ref_thumbnail_index가 우선된다.

<559> 단계 S43에서, 사용자는 재생 개시점의 마크점을 지정한다. 이 지정은, 예를 들면, 단계 S42의 처리에서 표시된 메뉴 화면상 중에서 사용자가 썸네일 화상을 선택하고, 그 썸네일에 대응되어 있는 마크점을 지정함으로써 행해진다.

<560> 단계 S44에서, 제어부(23)는 단계 S43의 처리에서 지정된 마크점의 RSPN_ref_EP_start와 offset_num_pictures(도 82)를 취득한다.

<561> 단계 S45에서, 제어부(23)는 단계 S44에서 취득한 RSPN_ref_EP_start에 대응하는 소스 패킷 번호로부터 전송 스트림의 데이터를 판독하고, AV 디코더(27)에 공급시킨다.

<562> 단계 S46에서, 제어부(23)는 AV 디코더(27)를 제어하고, RSPN_ref_EP_start로 참조되는 픽쳐로부터(표시하지 않음) 표시해야 할 픽쳐를 카운트 업하면서 카운트 값이 offset_num_pictures가 되었을 때, 그 픽쳐로부터 표시를 개시시킨다.

<563> 이상의 처리를 도 113 내지 도 115를 참조하여 더 설명한다. 이 예에 있어서는, DVR 전송 스트림 파일은 소스 패킷 번호 A로부터 장면이 개시되고 있고, 소스 패킷 번호 B로부터 소스 패킷 C까지 CM이 삽입되어 있다. 이 때문에, 도 114에 도시한 바와 같이, EP_map에는 RSPN_EP_start로서의 A, B, C에 대응하여, PTS_EP_start로서 PTS(A), PTS(B), PTS(C)가 등록되어 있다.

<564> 또한, 도 115에 도시한 바와 같이, 장면 개시, CM 개시 및 CM 종료의 마크 타입에 대응하여, mark_entry와 representative_picture_entry가 등록되어 있다. mark_entry에는 장면 개시, CM 개시 및 CM 종료에 대응하여, RSPN_ref_EP_start로서 각각 A, B, C가 등록되고, offset_num_pictures로서 M1, N1, N2가 등록되어 있다. 마찬가지로, representative_picture_entry에는 RSPN_ref_EP_start로서 장면 개시, CM 개시 및 CM 종료에 대응하여, 각각 A, B, C가 등록되고, offset_num_pictures로서 M2, N1, N2가 각각 등록되어 있다.

<565> 장면 개시에 해당하는 픽쳐로부터 검색하여 재생이 명령된 경우, 패킷 번호 A의 데이터로부터 시작되는 스트림으로부터 디코드가 개시되고, PTS(A)의 픽쳐로부터(표시하지 않음) 표시해야 할 픽쳐를 카운트 업하면서 offset_num_pictures가 M1의 값이 되었을 때, 그 픽쳐로부터 표시가 개시된다.

<566> 또한, mark_entry와 representative_picture_entry의 신택스가 도 82에 나타내는 구성인 경우에 있어서의 CM 스킵 재생의 처리에 대하여, 도 116의 흐름도를 참조하여 설명한다.

<567> 단계 S61에서, 제어부(23)는 EP_map(도 70), STC_Info(도 52), Program_Info(도 54) 및 ClipMark(도 78)의 정보를 취득한다.

<568> 단계 S62에서, 사용자가 CM 스킵 재생을 명령하면, 단계 S63에서, 제어부(23)는 마크 타입(도 79)이 CM 개시점

과 CM 종료점인 각 점의 마크 정보로서 RSPN_ref_EP_start와 offset_num_pictures(도 82)를 취득한다. 그리고, CM 개시점의 데이터는 RSPN_ref_EP_start(1), offset_num_pictures(1)로 되고, CM 종료점의 데이터는 RSPN_ref_EP_start(2), offset_num_pictures(2)로 된다.

<569> 단계 S64에서, 제어부(23)는 RSPN_ref_EP_start(1), RSPN_ref_EP_start(2)에 대응하는 PTS를 EP_map(도 70)로부터 취득한다.

<570> 단계 S65에서, 제어부(23)는 전송 스트림을 기록 매체(100)로부터 판독하여, AV 디코더(27)에 공급시킨다.

<571> 단계 S66에서, 제어부(23)는 현재의 표시 화상이 RSPN_ref_EP_start(1)에 대응하는 PTS의 픽처인지의 여부를 판정하고, 현재의 표시 화상이 RSPN_ref_EP_start(1)에 대응하는 PTS의 픽처가 아닌 경우에는, 단계 S67로 진행하여 픽처를 그대로 계속적으로 표시시킨다. 그 후, 처리는 단계 S65로 되돌아가서, 그 이후의 처리가 반복되어 실행된다.

<572> 단계 S66에서, 현재의 표시 화상이 RSPN_ref_EP_start(1)에 대응하는 PTS의 픽처라고 판정된 경우, 단계 S68로 진행하여 제어부(23)는 AV 디코더(27)를 제어하고, RSPN_ref_EP_start(1)에 대응하는 PTS의 픽처로부터 표시하는 픽처를 카운트 업하면서 카운트 값이 offset_num_pictures(1)가 되었을 때, 표시를 정지시킨다.

<573> 단계 S69에서, 제어부(23)는 RSPN_ref_EP_start(2)의 소스 패킷 번호로부터 전송 스트림의 데이터를 판독하여, AV 디코더(27)에 공급한다.

<574> 단계 S70에서, 제어부(23)는 AV 디코더(27)를 제어하고, RSPN_ref_EP_start(2)에 대응하는 PTS의 픽처로부터 (표시하지 않음) 표시해야 할 픽처를 카운트 업하면서 카운트 값이 offset_num_pictures(2)로 되었을 때, 그 픽처로부터 표시를 개시시킨다.

<575> 이상의 동작을 도 113 내지 도 115를 참조하여 다시 설명하면, 우선, EP_map(도 114)를 기초로, 패킷 번호 B, C에 대응하는 시각 PTS(B), PTS(C)가 얻어진다. 그리고, Clip AV stream이 디코드되어 가면서, 표시 시각이 PTS(B)로 되었을 때, PTS(B)의 픽처로부터 표시 픽처가 카운트 업되어 그 값이 N1(도 115)로 되었을 때, 표시가 정지된다.

<576> 또한, 패킷 번호 C의 데이터로부터 시작되는 스트림으로부터 디코드가 재개되고, PTS(C)의 픽처로부터(표시하지 않음) 표시해야 할 픽처를 카운트 업하면서 그 값이 N2(도 115)로 되었을 때, 그 픽처로부터 표시가 재개된다.

<577> 이상의 처리는 CM 스킵 재생에 한하지 않고, ClipMark로 지정된 2점 사이의 장면을 스킵시켜 재생하는 경우에도 적용 가능하다.

<578> 다음으로, mark_entry와 representative_picture_entry의 신택스가, 도 84에 도시한 바와 같은 구성인 경우에 있어서의 마크점으로 나타내는 장면의 재생 개시 처리에 대하여, 도 118의 흐름도를 참조하여 설명한다.

<579> 단계 S81에서, EP_map(도 70), STC_Info(도 52), Program_Info(도 54), 및 ClipMark(도 78)의 정보가 취득된다.

<580> 단계 S82에서, 제어부(23)는 ClipMark(도 78)의 representative_picture_entry 또는 ref_thumbnail_index로 참조되는 픽처로부터 썸네일의 리스트를 생성하고, GUI의 메뉴 화면으로서 표시시킨다. ref_thumbnail_index가 유효한 값을 갖는 경우, representative_picture_entry보다 ref_thumbnail_index가 우선된다.

<581> 단계 S83에서, 사용자는 재생 개시점의 마크점을 지정한다. 이 지정은, 예를 들면, 메뉴 화면상 중에서 사용자가 썸네일 화상을 선택하고, 그 썸네일에 대응되어 있는 마크점을 지정함으로써 행해진다.

<582> 단계 S84에서, 제어부(23)는 사용자로부터 지정된 mark_entry의 RSPN_mark_point(도 84)를 취득한다.

<583> 단계 S85에서, 제어부(23)는 마크점의 RSPN_mark_point보다 앞에 있으며, 또한, 가장 가까운 엔트리 포인트의 소스 패킷 번호를 EP_map(도 70)로부터 취득한다.

<584> 단계 S86에서, 제어부(23)는 단계 S85에서 취득한 엔트리 포인트에 대응하는 소스 패킷 번호로부터 전송 스트림의 데이터를 판독하여, AV 디코더(27)에 공급시킨다.

<585> 단계 S87에서, 제어부(23)는 AV 디코더(27)를 제어하고, RSPN_mark_point로 참조되는 픽처로부터 표시를 개시시킨다.

<586> 이상의 처리를 도 119 내지 도 121을 참조하여 더 설명한다. 이 예에 있어서는, DVR 전송 스트림 파일이 소스 패킷 A에서 장면 개시하고, 소스 패킷 번호 B로부터 C까지 CM이 삽입되어 있다. 이 때문에, 도 120의 EP_map에는 RSPN_EP_start로서의 A, B, C에 대응하여, PTS_EP_start가 각각 PTS(A), PTS(B), PTS(C)로서 등록되어 있다. 또한, 도 121에 나타내는 ClipMark에, 장면 개시, CM 개시 및 CM 종료에 대응하여, markentry의 RSPN_mark_point로서 a1, b1, c1이 등록되고, 또한, representative_picture_entry의 RSPN_mark_point로서 a2, b1, c1이 등록되어 있다.

<587> 장면 개시에 해당하는 픽쳐로부터 재생 개시하는 경우, 패킷 번호 A<a1로 하면, 패킷 번호 A의 데이터로부터 시작되는 스트림으로부터 디코드가 개시되고, 소스 패킷 번호 a1에 대응하는 픽쳐로부터 표시가 개시된다.

<588> 다음으로, mark_entry와 representative_picture_entry의 신택스가, 도 84에 나타내는 바와 같은 구성인 경우에 있어서의 CM 스킵 재생의 처리에 대하여, 도 122와 도 123의 흐름도를 참조하여 설명한다.

<589> 단계 S101에서, 제어부(23)는 EP_map(도 70), STC_Info(도 52), Program_Info(도 54) 및 ClipMark(도 70)의 정보를 취득한다.

<590> 단계 S102에서, 사용자는 CM 스킵 재생을 지정한다.

<591> 단계 S103에서, 제어부(23)는 마크 타입(도 79)이 CM 개시점과 CM 종료점인 각 점의 마크 정보의 RSPN_mark_point(도 84)를 취득한다. 그리고, 제어부(23)는 CM 개시점의 데이터를 RSPN_mark_point(1)로 하고, CM 종료점의 데이터를 RSPN_mark_point(2)로 한다.

<592> 단계 S104에서, 제어부(23)는 기록 매체(100)로부터 전송 스트림을 판독하고, AV 디코더(27)로 출력하여 디코드시킨다.

<593> 단계 S105에서, 제어부(23)는 현재의 표시 화상이 RSPN_mark_point(1)에 대응하는 픽쳐인지의 여부를 판정하고, 현재의 표시 화상이 RSPN_mark_point(1)에 대응하는 픽쳐가 아닌 경우에는, 단계 S106으로 진행하여 그대로 픽쳐를 계속적으로 표시시킨다. 그 후, 처리는 단계 S104로 되돌아가, 그 이후의 처리가 반복하여 실행된다.

<594> 단계 S105에서, 현재의 표시 화상이 RSPN_mark_point(1)에 대응하는 픽쳐라고 판정된 경우, 단계 S107로 진행하여, 제어부(23)는 AV 디코더(27)를 제어하여 디코드 및 표시를 정지시킨다.

<595> 다음으로, 단계 S108에서, RSPN_mark_point(2)보다 앞에 있으며, 또한, 가장 가까운 엔트리 포인트가 있는 소스 패킷 번호가 EP_map(도 70)으로부터 취득된다.

<596> 단계 S109에서, 제어부(23)는 단계 S108에서 취득한 엔트리 포인트에 대응하는 소스 패킷 번호로부터 전송 스트림의 데이터를 판독하여, AV 디코더(27)에 공급시킨다.

<597> 단계 S110에서, 제어부(23)는 AV 디코더(27)를 제어하고, RSPN_mark_point(2)로 참조되는 픽쳐로부터 표시를 재개시킨다.

<598> 이상의 처리를 도 119 내지 도 121의 예로 더 설명하면, Clip AV stream 을 디코드하여 가면서, 소스 패킷 번호 b1(도 121)에 대응하는 표시 픽쳐로 되었을 때, 표시가 정지된다. 그리고, 소스 패킷 번호 C<소스 패킷 번호 c1로 하면, 패킷 번호 C의 데이터로부터 시작되는 스트림으로부터 디코드가 재개되어, 소스 패킷 번호 c1에 대응하는 픽쳐로 되었을 때, 그 픽쳐로부터 표시가 재개된다.

<599> 이상과 같이 하여, 도 124에 도시한 바와 같이, PlayList 상에서 타임 스탬프에 의해 소정의 위치를 지정하고, 이 타임 스탬프를 각 Clip의 Clip Information에 있어서 데이터 어드레스로 변환하여, Clip AV stream의 소정 위치에 액세스할 수 있다.

<600> 보다 구체적으로는, 도 125에 도시한 바와 같이, PlayList 상에서, PlayListMark로서 북 마크나 리즘점을 사용자가 시간축 상의 타임 스탬프로서 지정하면, 그 PlayList는 재생될 때, 그 PlayList가 참조하고 있는 Clip의 ClipMark를 사용하여 Clip AV stream의 장면 개시점이나 장면 종료점에 액세스할 수 있다.

<601> 또, ClipMark의 신택스는 도 78의 예 대신에, 도 126에 도시한 바와 같이 할 수도 있다.

<602> 이 예에 있어서는, RSPN_mark가 도 78의 reserved_for_MakerID, mark_entry() 및 representative_picture_entry() 대신에 삽입되어 있다. 이 RSPN_mark의 32비트의 필드는, AV 스트림 파일 상에서, 그 마크가 참조하는 액세스 유닛의 제1 바이트째를 포함하는 소스 패킷의 상대 어드레스를 나타낸다. RSPN_mark는 소스 패킷 번호를 단위로 하는 크기로, AV 스트림 파일의 최초의 소스 패킷으로부터 Clip

Information file로 정의되며, offset_SPN의 값을 초기 값으로 하여 카운트된다.

<603> 그 밖의 구성은 도 78에서의 경우와 마찬가지이다.

<604> ClipMark의 신택스는 또한 도 127에 도시한 바와 같이 구성할 수도 있다. 이 예에 있어서는, 도 126에 있어서의 RSPN_mark 대신에, RSPN_ref_EP_start와 offset_num_pictures가 삽입되어 있다. 이들은 도 82에 도시한 경우와 마찬가지의 것이다.

<605> 도 128은 아날로그 AV 신호를 인코드하여 기록하는 경우, 도 81에 도시한 신택스의 ClipMark의 작성에 대하여 설명하는 흐름도이다. 도 1의 기록 재생 장치(1)의 블록도를 참조하면서 설명한다. 단계 S200에서, 해석부(14)는 단자(11, 12)로부터의 입력 AV 신호를 해석하여 특징점을 검출한다. 특징점은, AV 스트림의 내용에 기인하는 특징적인 장면을 지정하고, 예를 들면, 프로그램의 검색점이나 장면 전환점 등이다.

<606> 단계 S201에서, 제어부(23)는 특징점의 화상의 PTS를 취득한다. 단계 S202에서, 제어부(23)는 특징점의 정보를 ClipMark에 저장한다. 구체적으로는, 본 예의 ClipMark의 신택스와 시맨틱스로 설명한 정보를 저장한다. 단계 S203에서, Clip Information file과 Clip AV stream file이 디스크에 기록된다.

<607> 도 129는 디지털 인터페이스로부터 입력된 트랜스포트 스트림을 기록하는 경우, 도 81에 도시한 신택스의 ClipMark의 작성에 대하여 설명하는 흐름도이다. 도 1의 기록 재생 장치(1)의 블록도를 참조하면서 설명한다. 단계 S211에서, 디멀티플렉서(26) 및, 제어부(23)는 기록하는 프로그램의 엘리먼트리 스트림 PID를 취득한다. 해석 대상의 엘리먼트리 스트림이 복수개 있는 경우, 모든 엘리먼트리 스트림 PID가 취득된다.

<608> 단계 S212에서, 디멀티플렉서(26)는 단자(13)로부터 입력되는 트랜스포트 스트림의 프로그램으로부터 엘리먼트리 스트림을 분리하여, 그것을 AV 디코더(27)가 AV 신호로 디코드한다. 단계 S213에서, 해석부(14)는 상기 AV 신호를 해석하여 특징점을 검출한다.

<609> 단계 S214에서, 제어부(23)는 특징점의 화상의 PTS와, 그것이 속하는 STC의 STC-sequence-id를 취득한다. 단계 S215에서, 제어부(23)는 특징점의 정보를 ClipMark에 저장한다. 구체적으로는, 본 예에서의 ClipMark의 신택스와 시맨틱스로 설명한 정보를 저장한다.

<610> 단계 S216에서, Clip Information file과 Clip AV stream file가 디스크에 기록된다.

<611> 도 128에 도시한 흐름도, 및 도 129에 도시한 흐름도와 같이 하여, AV 스트림 파일, 즉 Clip AV 스트림 파일의 중의 특징적인 화상을 지시하는 마크를 저장한다. lipMark는 상기 AV 스트림의 관리 정보 데이터 파일, 즉 Clip Information 파일에 기록된다.

<612> 도 130은 Real PlayList의 작성에 대하여 설명하는 흐름도이다. 도 1의 기록 재생 장치(1)의 블록도를 참조하면서 설명한다. 단계 S221에서, 제어부(23)는 ClipAV 스트림을 기록한다. 단계 S222에서, 제어부(23)는 상기 Clip의 모든 재생 가능 범위를 커버하는 PlayItem으로 이루어지는 PlayList()를 작성한다. Clip 중에 STC 불연속점이 있고, PlayList()가 2개 이상의 PlayItem으로 이루어지는 경우, PlayItem 사이의 connection-condition 도 또한 결정된다.

<613> 단계 S223에서, 제어부(23)는, UIAppInfoPlayList()를 작성한다. 단계 S224에서, 제어부(23)는 PlayListMark를 작성한다. 단계 S225에서, 제어부(23)는 MakersPrivateData를 작성한다. 단계 S226에서, 제어부(23)는 Real PlayList 파일을 기록한다.

<614> 이와 같이 하여, 신규로 Clip AV 스트림을 기록 할 때마다, 하나의 Real Pl ayList 파일이 만들어진다.

<615> 도 131은 Virtual PlayList의 작성에 대하여 설명하는 흐름도이다. 단계 S231에서, 사용자 인터페이스를 통해 디스크에 기록되어 있는 하나의 Real PlayList의 재생이 지정된다. 그리고, 그 Real PlayList의 재생 범위 중에서, 사용자 인터페이스를 통해 IN점과 OUT점으로 표시되는 재생 구간이 지정된다.

<616> 단계 S232에서, 제어부(23)는 사용자에 의한 재생 범위의 지정 조작이 전부 종료하였는지의 여부를 판단한다. 단계 S232에서, 사용자에 의한 재생 범위의 지정 조작은 아직 종료하지 않았다고 판단된 경우, 단계 S231로 되돌아가, 그 이후의 처리가 반복되고, 종료하였다고 판단된 경우, 단계 S233으로 진행한다.

<617> 단계 S233에서, 연속하여 재생되는 2개의 재생 구간 사이의 접속 상태(connection_condition)는, 사용자가 사용자 인터페이스를 통해 결정되거나, 또는 제어부(23)에 의해 결정된다. 단계 S234에서, 사용자 인터페이스를 통해 사용자가 서브 패스(후기록용 오디오) 정보를 지정한다. 사용자가 서브 패스를 작성하지 않은 경우, 단계

S234에서의 처리는 스킵된다.

<618> 단계 S235에서, 제어부(23)는 사용자가 지정한 재생 범위 정보, 및 cbnnection_condition에 기초하여, PlayList()를 작성한다. 단계 S236에서, 제어부(23)는 UIAppInfoPlayList()를 작성한다. 단계 S237에서, 제어부(23)는 PlayListMark를 작성한다. 단계 S238에서, 제어부(23)는 MakersPrivateData를 작성한다. 단계 S239에서, 제어부(23)는 Virtual PlayList 파일을, 디스크에 기록시킨다.

<619> 이와 같이 하여, 디스크에 기록되어 있는 Real PlayList의 재생 범위 중에서, 사용자가 보고 싶은 재생 구간을 선택하여, 그 재생 구간을 그룹화할 때마다, 하나의 Virtual PlayList 파일이 작성된다.

<620> 도 132는 PlayList의 재생에 대하여 설명하는 흐름도이다. 도 1의 기록 재생 장치(1)의 블록도를 참조하면서 설명한다. 단계 S241에서, 제어부(23)는 Info.dvr, Clip Information file, PlayList file 및 썸네일 파일의 정보를 취득하고, 디스크에 기록되어 있는 PlayList의 일람을 나타내는 GUI 화면을 작성하여, 사용자 인터페이스를 통해 GUI에 표시한다.

<621> 단계 S242에서, 사용자 인터페이스를 통해 사용자가 하나의 PlayList의 재생을 제어부(23)에 지시한다. 단계 S243에서, 제어부(23)는 현재의 PlayItem의 STC-sequence-id와 IN_time의 PTS로부터, IN_time보다 시간적으로 앞에서 가장 가까운 엔트리 포인트가 있는 소스 패킷 번호를 취득한다. 단계 S244에서, 제어부(23)는 상기 엔트리 포인트가 있는 소스 패킷 번호로부터 AV 스트림의 데이터를 판독하여, AV 디코더(27)로 공급한다.

<622> 상기 PlayItem의 시간적으로 앞에서 PlayItem의 재생이 있었던 경우, 단계 S245에서, 제어부(23)는 그 PlayItem과의 표시의 접속 처리를 connection_condition에 따라 행해지도록 제어를 행한다. 단계 S246에서, AV 디코더(27)는 IN_time의 PTS의 픽쳐로부터 표시를 개시한다.

<623> 단계 S247에서, AV 디코더(27)는 AV 스트림의 디코드를 계속적으로 행한다. 단계 S248에서, 제어부(23)는 현재 표시의 화상이 OUT_time의 PTS의 화상인지의 여부를 판단한다. 단계 S248에서, 현재 표시의 화상은 OUT_time의 PTS의 화상이라고 판단된 경우, 단계 S250으로 진행하고, PTS의 화상이 아니라고 판단된 경우, 단계 S249로 진행한다.

<624> 단계 S249에서, PTS의 화상이라고 판단된 화상을 표시하기 위한 처리가 실행되고, 그 후 단계 S247로 되돌아가, 그 이후의 처리가 반복된다. 한편, 단계 S250에서는, 제어부(23)에 의해 현재의 Play Item이 PlayList 중에서 최후의 PlayItem인지의 여부가 판단된다. 단계 S250에서, 현재의 PlayItem이 PlayList 중에서 최후의 PlayItem이라고 판단된 경우, 도 132에 도시한 흐름도의 처리는 종료되고, 최후의 PlayItem이 아니라고 판단된 경우, 단계 S243으로 되돌아가, 그 이후의 처리가 반복된다.

<625> 도 133은, PlayListMark의 작성에 대하여 설명하는 흐름도이다. 도 1의 기록 재생 장치(1)의 블록도를 참조하면서 설명한다. 단계 S261에서, 제어부(23)는 Info.dvr, Clip Information file, PlayList file 및 Thumbnail file의 정보를 취득하고, 디스크에 기록되어 있는 PlayList의 일람을 나타내는 GUI 화면을 작성하여, 사용자 인터페이스를 통해 GUI에 표시한다.

<626> 단계 S262에서, 사용자 인터페이스를 통해 사용자에 의해 하나의 PlayList의 재생이 제어부(23)에 지시된다. 단계 S263에서, 재생부(3)는 지시된 PlayList의 재생을 개시한다 (도 132의 흐름도를 참조하여 설명한 바와 같이 행해진다).

<627> 단계 S264에서, 사용자 인터페이스를 통해 사용자에 의해, 마음에 드는 장면의 부분에 마크의 세트가 제어부(23)에 지시된다. 단계 S265에서, 제어부(23)는 마크의 PTS와, 그것이 속하는 PlayItem의 PlayItem_id를 취득한다.

<628> 단계 S266에서, 제어부(23)는 마크의 정보를 PlayListMark()에 저장한다. 단계 S267에서, PlayList 파일이 디스크에 기록된다.

<629> 이와 같이 하여, PlayList의 재생 범위 중에서 사용자가 지정한 마크점, 또는, 그 PlayList를 재생할 때의 리즘점을 나타내는 마크를 저장하는 PlayList Nark를 PlayList 파일에 기록된다.

<630> 도 134는 PlayList가 재생될 때, PlayListMark 및 그 PlayList가 참조하는 Clip의 ClipMark가 사용된 프로그램 시작 검출 위치 재생에 대하여 설명하는 흐름도이다. ClipMark()의 신택스는, 도 81에 도시하도록 한다. 도 1의 기록 재생 장치(1)의 블록도를 참조하면서 설명한다.

<631> 단계 S271에서, 제어부(23)는 Info.dvr, Clip Information file, PlayList file 및 Thumbnail file의 정보를

취득하고, 디스크에 기록되어 있는 PlayList의 일람을 나타내는 GUI 화면을 작성하여, 사용자 인터페이스를 통해 GUI에 표시한다.

<632> 단계 S272에서, 사용자 인터페이스를 통해 사용자에 의해 하나의 PlayList의 재생이 지시된다. 단계 S273에서, 제어부(23)는 PlayListMark와 그 PlayList가 참조하는 Clip의 ClipMark로 참조되는 픽쳐로부터 생성한 썸네일의 리스트를, 사용자 인터페이스를 통해 GUI에 표시한다.

<633> 단계 S274에서, 사용자 인터페이스를 통해 제어부(23)에, 사용자에 의해 재생 개시점의 마크점이 지정된다. 단계 S275에서, 제어부(23)는 단계 S274에서의 처리로 선택된 마크가 PlayListMark에 저장되어 있는 마크인지의 여부를 판단한다. 단계 S275에서, 선택된 마크가 PlayListMark에 저장되어 있는 마크라고 판단된 경우, 단계 S276으로 진행하고, 저장되어 있지 않은 마크라고 판단된 경우, 단계 S278로 진행한다.

<634> 단계 S276에서, 제어부(23)는 마크의 PTS와, 그것이 속하는 PlayItem_id를 취득한다. 단계 S277에서, 제어부(23)는 PlayItem_id가 가리키는 PlayItem이 참조하는 AV 스트림의 STC-sequence-id를 취득한다.

<635> 단계 S278에서, 제어부(23)는 STC-sequence-id와 마크의 PTS에 기초하여, AV 스트림을 AV 디코더(27)로 입력시킨다. 구체적으로는, 이 STC-sequence-id와 마크점의 PTS를 이용하여, 도 132의 흐름도의 단계 S243, S244와 마찬가지의 처리가 행해진다. 단계 S279에서, 재생부(3)는 마크점의 PTS의 픽쳐로부터 표시를 개시한다.

<636> 도 9를 참조하여 설명한 바와 같이, PlayList가 재생될 때, 그 PlayList가 참조하는 Clip의 ClipMark에 저장되어 있는 마크를 참조할 수 있다. 따라서, 하나의 Clip를, Real PlayList나 복수의 Virtual PlayList에 의해서 참조하고 있는 경우, 이들의 PlayList는, 그 하나의 Clip의 ClipMark를 공유할 수 있기 때문에, 마크의 데이터를 효율적으로 관리할 수 있다.

<637> 만일, Clip에 ClipMark를 정의하지 않고, PlayList만으로 PlayListMark와 ClipMark를 정합한 것을 정의하도록 한 경우, 상기한 예와 같이 하나의 Clip을 Re al PlayList나 복수의 Virtual PlayList에 의해 참조하고 있는 경우, 각각의 PlayList가 동일한 내용의 Clip의 마크 정보를 갖음으로써, 데이터의 기록 효율이 나쁘다.

<638> 도 135는 PlayListMark()의 신택스의 다른 예를 나타내는 도면이다. length는, 이 length 필드의 직후의 바이트로부터 PlayListMark()의 최후의 바이트까지의 바이트 수를 나타낸다. number_of_PlayList_marks는, PlayListMark 중에 저장되어 있는 마크의 엔트리 수를 나타낸다.

<639> mark_invalid_flag는 1비트의 플래그로서, 이것의 값이 0으로 세트되어 있을 때, 이 마크는 유효한 정보를 가지고 있는 것을 나타내며, 또한, 이것의 값이 1로 세트되고 있을 때는, 이 마크는 무효한 것을 나타낸다.

<640> 사용자가 사용자 인터페이스 상에서 하나의 마크의 엔트리를 소거하는 조작을 했을 때, 기록 재생 장치(1)는 PlayListMark로 그 마크의 엔트리를 소거하는 대신에, 그 mark_invalid_flag의 값을 1로 변경하도록 하여도 무방하다.

<641> mark_type은 마크의 타입을 나타내고, 도 136에 도시한 의미를 갖는다. mark_name_length는, Mark_name 필드 중에 나타나는 마크명의 바이트 길이를 나타낸다. 이 필드의 값은 32 이하이다. ref_to_PlayItem_id는, 마크가 위치한 부분의 PlayItem을 지정하는 부분의 PlayItem_id의 값을 나타낸다. 임의의 PlayItem에 대응하는 PlayItem_id의 값은 PlayList()로 정의된다.

<642> mark_time_stamp는 그 마크가 지정된 포인트를 나타내는 타임 스탬프를 저장한다. mark_time_stamp는, ref_to_PlayItem_id로 나타나는 PlayItem 중에서 정의되어 있는 부분의 IN_time과 OUT_time으로 특정되는 재생 범위 내의 시간을 가리킨다. 타임스탬프의 의미는 도 44와 동일하다.

<643> entry_ES_PID가 0xFFFF로 세트되어 있는 경우, 그 마크는 PlayList에 의해서 사용되는 모든 엘리먼트리 스트림에 공통의 시간 축 상으로의 포인터이다. entry_ES_PID가 0xFFFF가 아닌 값으로 세트되어 있는 경우, entry_ES_PID는, 그 마크에 의해 가리키는 부분의 엘리먼트리 스트림을 포함하고 있는 부분의 트랜스포트 패킷의 PID의 값을 나타낸다.

<644> ref_thumbnail_index는, 마크에 부가되는 썸네일 화상의 정보를 나타낸다. 그 의미는, 도 42의 ref_Thumbnail_index와 동일하다. mark_name은, 마크의 이름을 나타낸다. 이 필드 중의 좌측으로부터 mark_name_length로 나타나는 바이트 수가, 유효한 캐릭터 문자로서, 이름을 나타낸다. 이 캐릭터 문자는, UIAppInfoPlayList 중에서 character_set에 의해 표시되는 방법으로 부호화되어 있다.

<645> mark_name 필드 중에서, 이들 유효한 캐릭터 문자에 계속되는 바이트의 값은 어떤 값이 들어가 있어도

무방하다. 이 신택스의 경우, 마크가 특정한 엘리먼트리 스트림을 가리킬 수 있다. 예를 들면, PlayList가 프로그램 중에 복수의 비디오 스트림을 갖는 멀티 뷰 프로그램을 참조하고 있을 때, entry_ES_PID는 그 프로그램의 중의 하나의 비디오 스트림을 나타내는 비디오 PID를 세트하기 위해 사용된다.

<646> 사용자가 멀티 뷰 프로그램을 참조하는 부분의 PlayList를 재생하고 있고, 그 사용자는 멀티 뷰 중의 하나의 뷰를 보고 있다고 상정한다. 지금, 사용자가 기록 재생 장치(1)에 대하여, 다음의 마크점에 재생을 스킵하도록 커맨드를 보내었다고 가정한다. 이 경우, 기록 재생 장치(1)는 사용자가 현재 보고 있는 뷰의 비디오 PID와 동일한 값인 부분의 entry_ES_PID의 마크를 사용해야 하고, 기록 재생 장치(1)는 멋대로 뷰를 변경해야만 하는 것은 아니다. 기록 재생 장치(1)는, 또한, entry_ES_PID가 0xFFFF로 세트되어 있는 마크를 사용하여도 된다. 이 경우에도 기록 재생 장치(1)는 마음대로 뷰를 변경하지 않는다.

<647> 도 137은, 도 81에 도시한 신택스의 ClipMark()의 다른 예를 나타내는 도면이다. length는, 이 length 필드의 직후의 바이트로부터 ClipMark()의 최후의 바이트까지의 바이트 수를 나타낸다. maker_ID는, mark_type이 0x60으로부터 0x7F의 값을 나타낼 때, 그 mark_type을 정의하고 있는 메이커의 메이커 ID를 나타낸다.

<648> number_of_Clip_marks는, ClipMark 중에 저장되어 있는 마크의 엔트리 수를 나타낸다. mark_invalid_flag는 1비트의 플래그로서, 이것의 값이 0으로 세트되어 있을 때, 이 마크는 유효한 정보를 갖고 있는 것을 나타내며, 또한, 이 값이 1로 세트되고 있을 때, 이 마크는 무효한 것을 나타낸다.

<649> 사용자가, 사용자 인터페이스 상에서 하나의 마크의 엔트리를 소거하는 조작을 했을 때, 기록기는 ClipMark로부터 그 마크의 엔트리를 소거하는 대신에, 그 mark_invalid_flag의 값이 1로 변경되도록 하여도 된다. mark_type은, 마크의 타입을 나타내며, 도 138에 도시한 의미를 갖는다 .

<650> ref_to_STC_id는, mark_time_stamp와 representative_picture_time_stamp가 모두 위치하고 있는 부분의 STC-sequence를 지정하는 부분의 STC-sequence-id를 나타낸다. STC-sequence-id의 값은, STCInfo() 중에서 정의된다. mark_time_stamp는, 도 81의 mark_entry()의 경우에서의 mark_time_stamp와 동일한 의미이다.

<651> entry_ES_PID가 0xFFFF로 세트되어 있는 경우, 그 마크는 Clip 중의 모든 엘리먼트리 스트림에 공통의 시간 축상으로의 포인터이다. 엔트리 S-PID가 0xFFFF가 아닌 값으로 세트되어 있는 경우, entry_ES_PID는 그 마크에 의해 지정되는 부분의 엘리먼트리 스트림을 포함하고 있는 부분의 트랜스포트 패킷의 PID의 값을 나타낸다.

<652> ref_to_thumbnail_index는, 마크에 부가되는 썸네일 화상의 정보를 나타낸다. 그 의미는, 도 78의 ref_thumbnail_index와 동일하다. representative_picture_time_stamp는, 도 81의 representative_picture_entry()의 경우에서의 mark_time_stamp와 동일한 의미이다.

<653> 도 137에 도시한 신택스의 경우, 마크가, 특정한 엘리먼트리 스트림을 가리킬 수 있다. 예를 들면, Clip이, 프로그램 중에 복수의 비디오 스트림을 갖는 멀티 뷰 프로그램을 포함하고 있을 때, entry_ES_PID는, 그 프로그램의 중의 하나의 비디오 스트림을 나타내는 비디오 PID를 세트하기 위해 사용된다.

<654> 사용자가 멀티 뷰 프로그램을 참조하는 부분의 PlayList를 재생하고 있고, 그 사용자는 멀티 뷰 중의 하나의 뷰를 보고 있다고 상정한다. 지금, 사용자가 기록 재생 장치(1)에 대하여, 다음의 마크점에 재생을 스킵하도록 커맨드를 보내었다가정한다. 이 경우, 기록 재생 장치(1)는 사용자가 현재 보고 있는 뷰의 비디오 PID와 동일한 값인 부분의 entry_ES_PID의 마크를 사용해야 하고, 기록 재생 장치(1)는, 멋대로 뷰를 변경해야 하는 것은 아니다. 기록 재생 장치(1)는 또한, entry_ES_PID가 0xFFFF로 세트되어 있는 마크를 사용해도 무방하다. 이 경우에도 기록 재생 장치(1)는 마음대로 뷰를 변경하지 않는다.

<655> 이러한 신택스, 데이터 구조, 규칙에 기초함으로써, 기록 매체(100)에 기록되어 있는 데이터의 내용, 재생 정보 등을 적절하게 관리할 수가 있기 때문에, 사용자가 재생 시에 적절하게 기록 매체에 기록되어 있는 데이터의 내용을 확인하거나, 원하는 데이터를 간편하게 재생할 수 있도록 할 수 있다.

<656> 이상과 같은 데이터 베이스 구성에 따르면, PlayList 파일이나 Clip Information 파일을 각각 분리하여 기록하기 때문에, 편집 등에 의해, 소정의 PlayList나 Clip의 내용이 변경되었을 때, 그 파일에 관계가 없는 다른 파일을 변경할 필요가 없다. 따라서, 파일의 내용의 변경이 용이하게 행할 수 있고, 또한 그 변경 및 기록에 소요되는 시간을 줄일 수 있다.

<657> 또한, 처음에 Info.dvr만을 판독하여, 디스크의 기록 내용을 사용자 인터페이스에 제시하고, 사용자가 재생 지시한 PlayList 파일과 그것에 관련하는 Clip Information 파일만을 디스크로부터 판독하도록 하면, 사용자의 대

기 시간을 줄일수 있다.

<658> 만일, 모든 PlayList 파일이나 Clip Information 파일을 하나의 파일에 통합하여 기록하면, 그 파일 사이즈는 매우 커진다. 그 때문에, 그 파일의 내용을 변경하고, 그것을 기록하기 위해 소요되는 시간은, 개개의 파일을 따로따로 분리하여 기록하는 경우에 비교하여, 매우 커진다. 그러나, 본 발명을 적용함으로써 이러한 것을 방지하는 것이 가능해진다.

<659> 상술한 바와 같이, AV 스트림 파일, 즉 Clip AV 스트림 파일의 중의 특징적인 화상을 지시하는 마크를 저장한다. ClipMark를, 상기 AV 스트림의 관리 정보 데이터 파일, 즉 Clip Information 파일에 기록하고, 또한, AV 스트림 중의 지정된 구간의 조합에 의해 정의되는 하나의 재생 수순의 정보를 갖는 오브젝트, 즉 PlayList의 재생 범위 중으로부터, 사용자가 지정한 마크점, 또는, 그 오브젝트를 재생할 때의 리즘점을 나타내는 마크를 저장하는 PlayListMark를, 오브젝트에 기록한다.

<660> 이와 같이 함으로써, PlayList가 재생될 때, 그 PlayList가 참조하는 Clip의 ClipMark에 저장되어 있는 마크를 참조할 수 있다. 따라서, 하나의 Clip을 Real PlayList나 복수의 Virtual PlayList에 의해 참조하고 있는 경우, 이들의 PlayList는 그 하나의 Clip의 ClipMark를 공유할 수 있기 때문에, 마크의 데이터를 효율적으로 관리할 수 있다.

<661> 만일, Clip에 ClipMark를 정의하지 않고, PlayList만으로 PlayListMark와 ClipMark를 정합한 것을 정의하도록 한 경우, 상기한 예와 같이 하나의 Clip을 Real PlayList나 복수의 Virtual PlayList에 의해 참조하고 있는 경우, 각각의 PlayList가 동일한 내용의 Clip의 마크 정보를 갖음으로써, 데이터의 기록 효율이 나쁘다. 그러나, 본 발명을 적용함으로써 이러한 것을 방지하는 것이 가능해진다.

<662> 이상과 같이, AV 스트림의 부속 정보로서, 엔트리 포인트의 어드레스를 저장하기 위한 EP_map과, 마크점의 픽처 타입(예를 들면 프로그램의 검색점)과 그 픽처의 AV 스트림 중의 어드레스를 저장하기 위한 ClipMark을, Clip Information File로서 파일화하여 기록 매체(100)에 기록함으로써, AV 스트림의 재생에 필요한 스트림의 재생에 필요한 스트림의 부호화 정보를 적절하게 관리하는 것이 가능하다.

<663> 이 Clip Information file 정보에 의해, 사용자가, 기록 매체(100)에 기록되어 있는 AV 스트림 중에서 흥미가 있는 장면, 예를 들면 프로그램의 검색점 등을 검색할 수가 있어, 사용자의 랜덤 액세스나 특수 재생의 지시에 대하여, 기록 매체(100)로부터의 AV 스트림의 판독 위치의 결정이 용이하게 되고, 또한 스트림의 복호 개시를 보다 빠르게 행할 수 있다.

<664> 상술한 일련의 처리는, 하드웨어에 의해 실행시키는 것도 가능하지만, 소프트웨어에 의해 실행시킬 수도 있다. 이 경우, 예를 들면, 기록 재생 장치(1)는 도 139에 도시한 바와 같은 퍼스널 컴퓨터에 의해 구성된다.

<665> 도 139에서, CPU(Central Processing Unit: 201)는 ROM(Read Only Memory: 202)에 기억되어 있는 프로그램, 또는 기억부(208)로부터 RAM(Random Access Memory: 203)에 로드된 프로그램에 따라 각종의 처리를 실행한다. RAM(203)에는, CPU(201)가 각종의 처리를 실행하기 위해 필요한 데이터 등도 적절하게 기억된다.

<666> CPU(201), ROM(202) 및 RAM(203)은, 버스(204)를 통해 서로 접속되어 있다. 이 버스(204)에는 또한 입출력 인터페이스(205)도 접속되어 있다.

<667> 입출력 인터페이스(205)에는, 키보드, 마우스 등으로 이루어지는 입력부(206), CRT, LCD 등으로 이루어지는 디스플레이, 및 스피커 등으로 이루어지는 출력부(207), 하드디스크 등으로 구성되는 기억부(208), 모뎀, 터미널 어댑터 등으로 구성되는 통신부(209)가 접속되어 있다. 통신부(209)는, 네트워크를 통한 통신 처리를 행한다.

<668> 또한, 입출력 인터페이스(205)에는 필요에 따라 드라이브(210)가 접속되고, 자기 디스크(221), 광 디스크(222), 광 자기 디스크(223), 혹은 반도체 메모리(224) 등이 적절하게 장착되고, 이들로부터 판독된 컴퓨터 프로그램이, 필요에 따라 기억부(208)에 인스톨된다.

<669> 상술한 일련의 처리는, 하드웨어에 의해 실행시킬 수도 있지만, 소프트웨어에 의해 실행시킬 수도 있다. 일련의 처리를 소프트웨어에 의해 실행시키는 경우에는, 그 소프트웨어를 구성하는 프로그램이 전용 하드웨어에 내장되어 있는 컴퓨터, 또는, 각종의 프로그램을 인스톨함으로써, 각종의 기능을 실행하는 것이 가능한, 예를 들면 범용의 퍼스널 컴퓨터 등에 기록 매체로부터 인스톨된다.

<670> 이 기록 매체는, 도 139에 도시한 바와 같이, 컴퓨터와는 별도로, 사용자에게 프로그램을 제공하기 위해 배포되는, 프로그램이 기록되어 있는 자기 디스크(221: 플로피 디스크를 포함함), 광 디스크(222)(CD-ROM(Compact

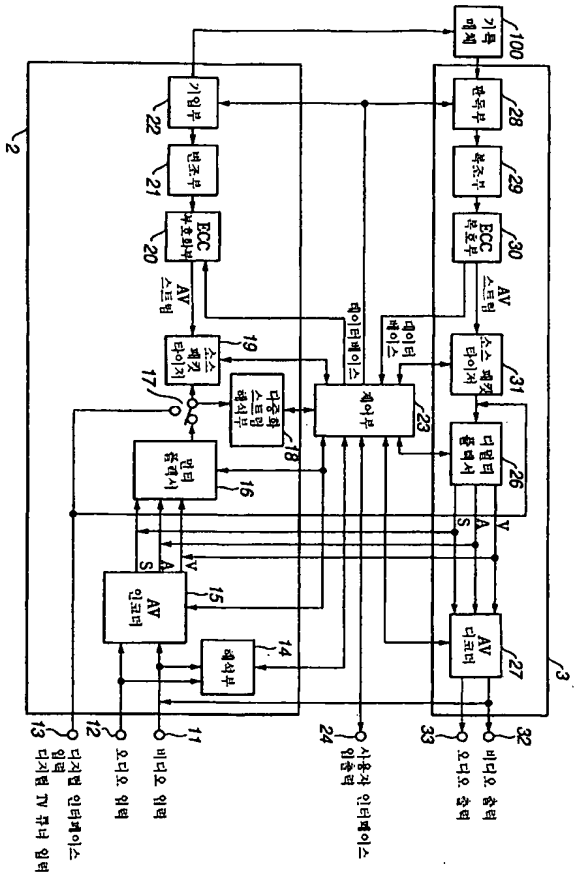
Disk-Read Only Memory), DVD(Digital Versatile Disk)를 포함함), 광 자기 디스크(223)(MD (Mini-Disk)를 포함함), 혹은 반도체 메모리(224) 등으로 이루어지는 패키지 미디어에 의해 구성되는 것뿐만 아니라, 컴퓨터에 사전에 내장된 상태에서 사용자에게 제공되는, 프로그램이 기억되어 있는 ROM(202)이나 기억부(208)가 포함되는 하드디스크 등으로 구성된다.

<671> 또, 본 명세서에서, 매체에 의해 제공되는 프로그램을 기술하는 단계는, 기재된 순서에 따라서, 시계열적으로 행해지는 처리는 물론, 반드시 시계열적으로 처리되지 않아도, 병렬적 혹은 개별로 실행되는 처리도 포함하는 것이다.

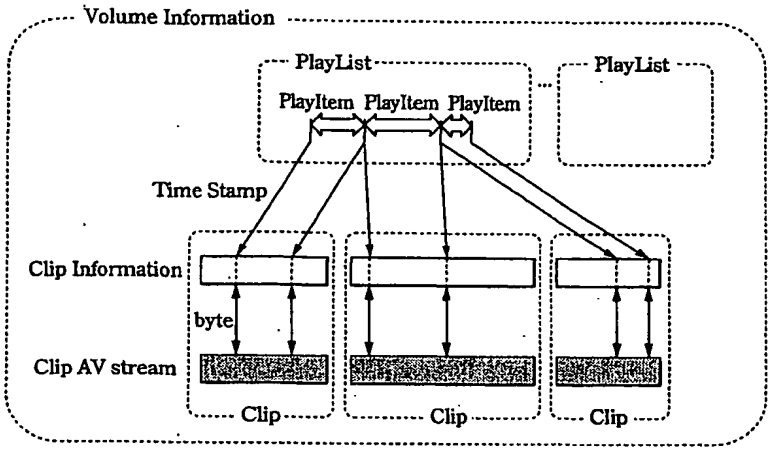
<672> 또한, 본 명세서에서, 시스템이란, 복수의 장치에 의해 구성되는 장치 전체를 나타내는 것이다.

도면

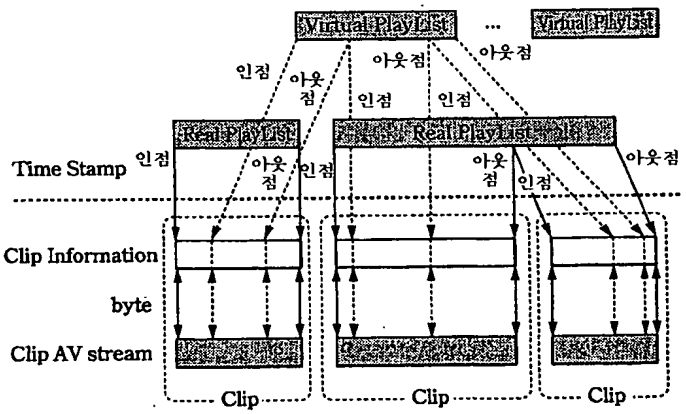
도면1



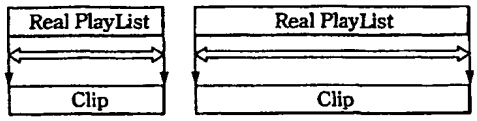
도면2


Volume Information
PlayList
PlayList
PlayItem
PlayItem
PlayItem
Time Stamp
Clip Information
byte
Clip AV stream
Clip
Clip
Clip

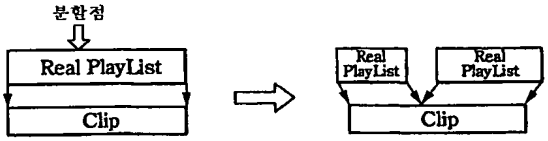
도면3


Virtual PlayList
Virtual PlayList
인점
아웃점
아웃점
인점
인점
Real PlayList
Real PlayList
Time Stamp
인점
아웃점
인점
아웃점
인점
아웃점
Clip Information
byte
Clip AV stream
Clip
Clip
Clip

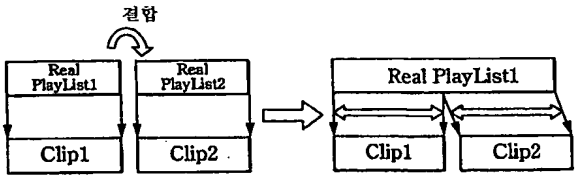
도면4


Real PlayList
Real PlayList
Clip
Clip

A

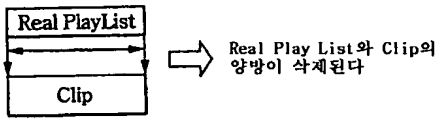

분할점
Real PlayList
Clip
Real PlayList
Real PlayList
Clip

B

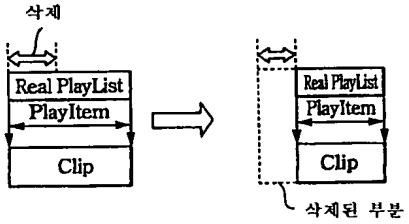

결합
Real PlayList1
Real PlayList2
Clip1
Clip2
Real PlayList1
Clip1
Clip2

C

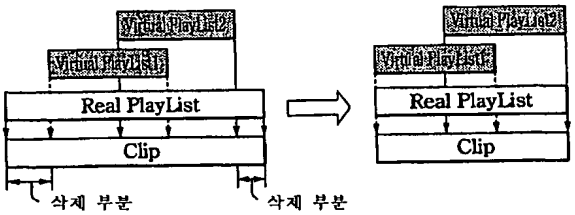
도면5


Real PlayList
Clip
Real Play List와 Clip의
양방이 삭제된다

A

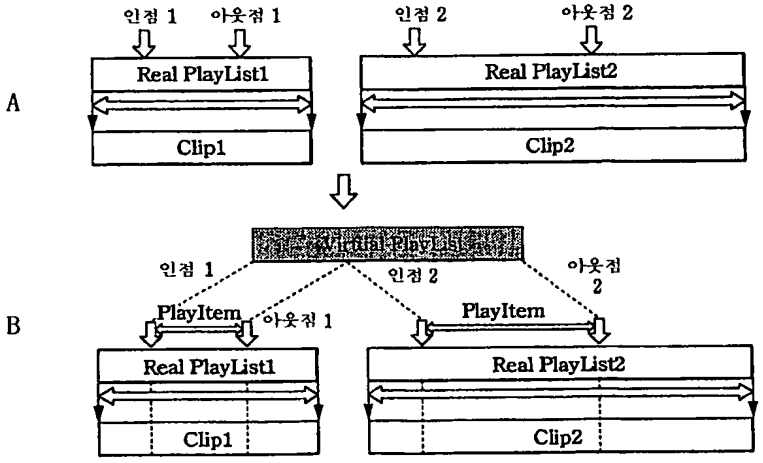

삭제
Real PlayList
PlayItem
Clip
Real PlayList
PlayItem
Clip
삭제된 부분

B

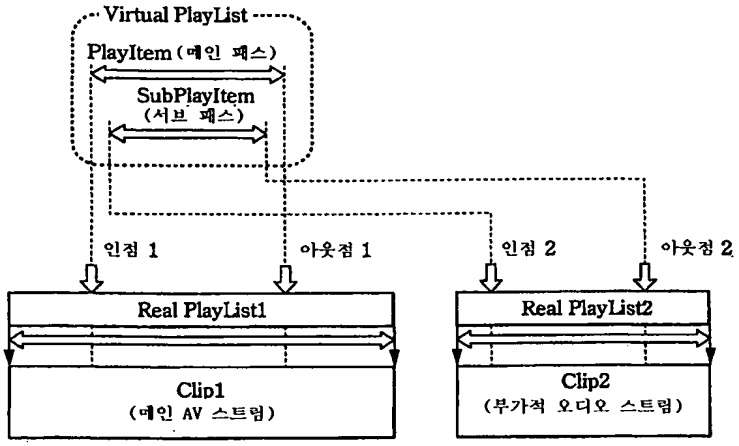

Real PlayList
Clip
삭제 부분
삭제 부분
Real PlayList
Clip

C

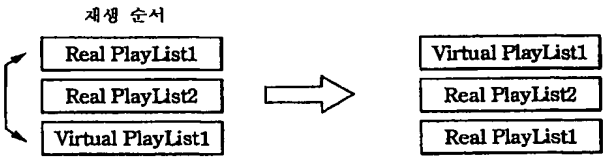
도면6


A
인점 1
아웃점 1
인점 2
아웃점 2
Real PlayList1
Clip1
Real PlayList2
Clip2
B
인점 1
인점 2
아웃점 2
PlayItem
아웃점 1
PlayItem
Real PlayList1
Clip1
Real PlayList2
Clip2

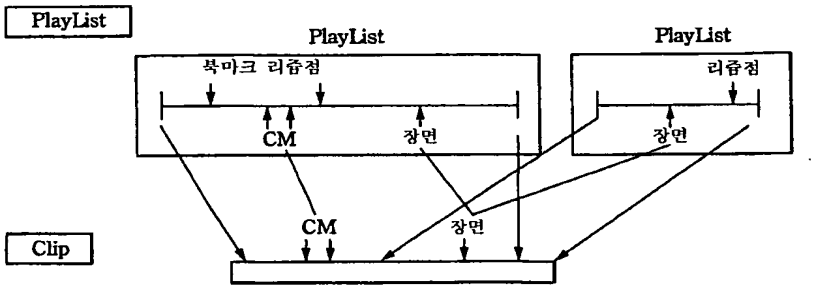
도면7


Virtual PlayList
PlayItem (메인 패스)
SubPlayItem
(서브 패스)
인점 1
아웃점 1
인점 2
아웃점 2
Real PlayList1
Real PlayList2
Clip1
(메인 AV 스트림)
Clip2
(부가적 오디오 스트림)

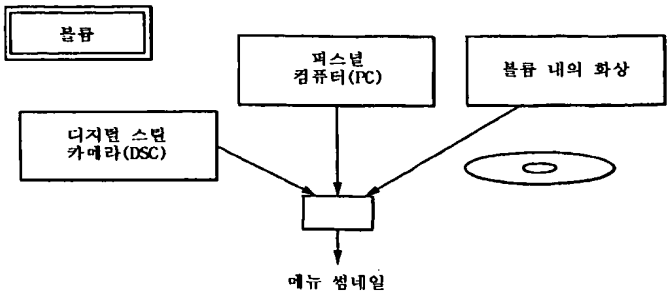
도면8


재생 순서
Real PlayList1
Real PlayList2
Virtual PlayList1
Virtual PlayList1
Real PlayList2
Real PlayList1

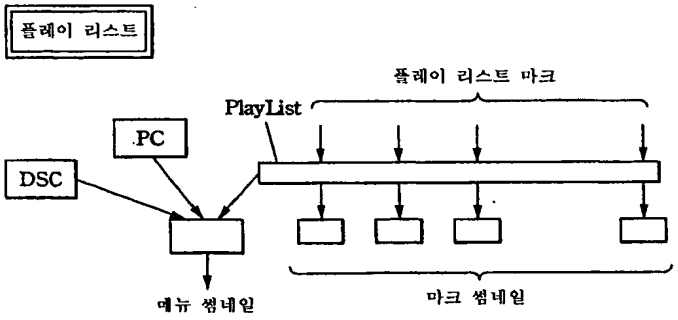
도면9


PlayList
PlayList
PlayList
북마크 리즘점
리즘점
CM
장면
장면
CM
장면
Clip

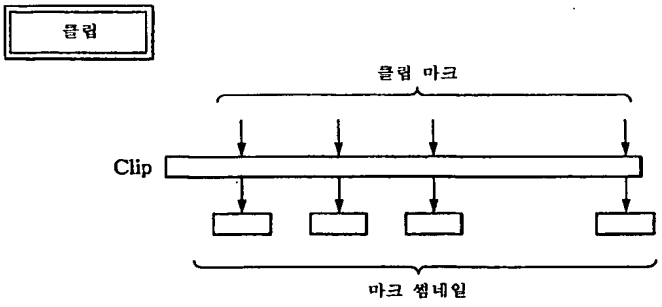
도면10



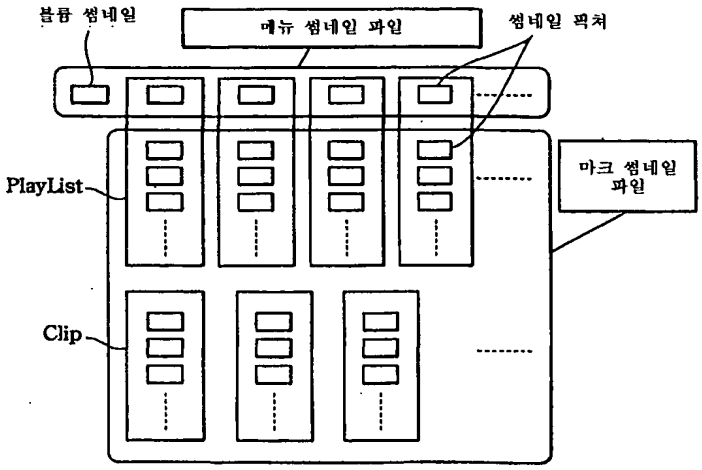
도면11



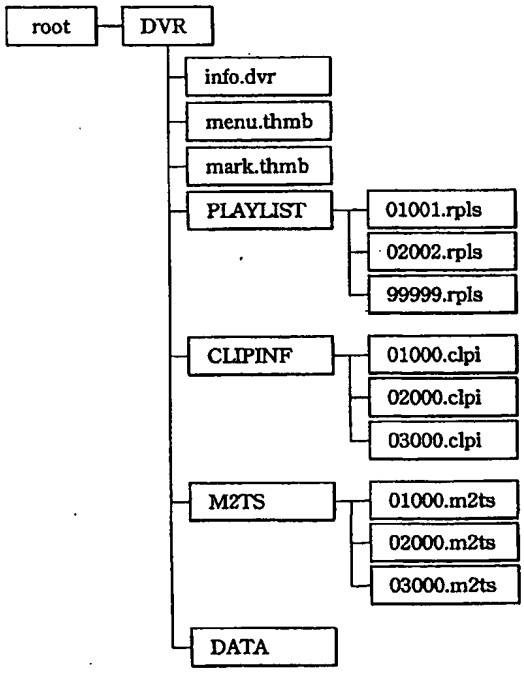
도면12



도면13


블럼 썸네일
메뉴 썸네일 파일
썸네일 픽처
PlayList
마크 썸네일 파일
Clip

도면14


root
DVR
info.dvr
menu.thmb
mark.thmb
PLAYLIST
01001.rpls
02002.rpls
99999.rpls
CLIPINF
01000.clpi
02000.clpi
03000.clpi
M2TS
01000.m2ts
02000.m2ts
03000.m2ts
DATA

도면15

신택스	바이트수	약호
info.dvr {		
TableOfPlayLists_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
DVRVolume()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

도면16

신택스	바이트수	약호
DVRVolume(){		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

도면17

신택스	바이트수	약호
ResumeVolume(){		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

도면18

신택스	바이트수	약호
UIAppInfoVolume(){		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

도면19

값	캐릭터 문자 부호화
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

도면20

신택스	바이트수	약호
TableOfPlayLists() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<*number_of_PlayLists*; i++) {		
PlayList_file_name	8*10	bslbf
}		
}		

도면21

신택스	바이트수	약호
TableOfPlayLists() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<*number_of_PlayLists*; i++) {		
PlayList_file_name	8*10	bslbf
UIAppInfoPlayList()		
}		
}		

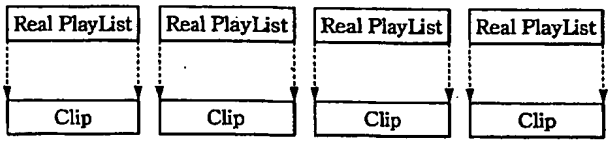
도면22

신택스	바이트수	약호
MakersPrivateData(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<*number_of_maker_entries*; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(j=0; j<*number_of_mpd_blocks*; j++){		
mpd_block	mpd_block_ size*1024*8	
}		
}		
}		

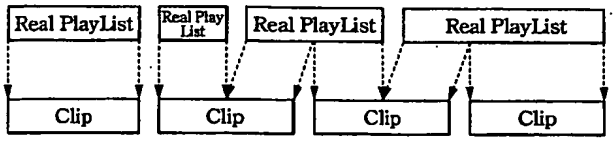
도면23

신택스	바이트수	약호
xxxxx.rpls / yyyyy.vpls {		
PlayListMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
PlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

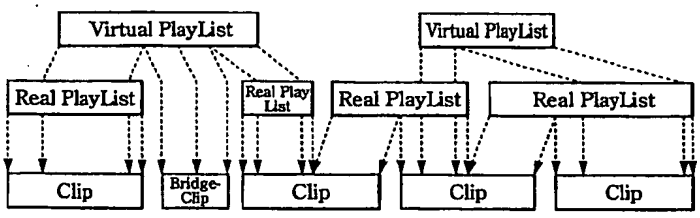
도면24


Real PlayList
Real PlayList
Real PlayList
Real PlayList
Clip
Clip
Clip
Clip

A


Real PlayList
Real Play List
Real PlayList
Real PlayList
Clip
Clip
Clip
Clip

B


Virtual PlayList
Virtual PlayList
Real PlayList
Real Play List
Real PlayList
Real PlayList
Clip
Bridge-Clip
Clip
Clip
Clip

C

도면25

신택스	바이트수	약호
PlayList(){		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	uimsbf
if (*<Vertual PlayList>*){		
number_of_SubPlayItems // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (*PlayItem_id=0;* *PlayItem_id<nymber_of_PlayItems;* *PlayItem_id++*){		
PlayItem() //main path		
}		
if (*<Virtual PlayList>*){		
if (CPI_type==0 && PlayList_type==0){		
for (j=0; i<*number_of _SubPlayItems*; i++)		
SubPlayItem() //sub path		
}		
}		
}		

도면26

PlayList_type	의미
0	AV 기록을 위한 PlayList 이 PlayList에 참조되는 모든 Clip는 하나 이상의 비디오 스트림을 포함해야 한다
1	오디오 기록을 위한 PlayList 이 PlayList에 참조되는 모든 Clip는 하나 이상의 오디오 스트림을 포함해야 하고, 그리고 비디오 스트림을 포함해서는 안된다.
2-255	reserved

도면27

신택스	바이트수	약호
UIAppInfoPlayList2() {		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

도면28

write_protect_flag	의미
0b	그 PlayList를 자유롭게 소거해도 된다
1b	write_protect_flag를 제외하고 그 PlayList의 내용은 소거 및 변경되어서는 안된다

A

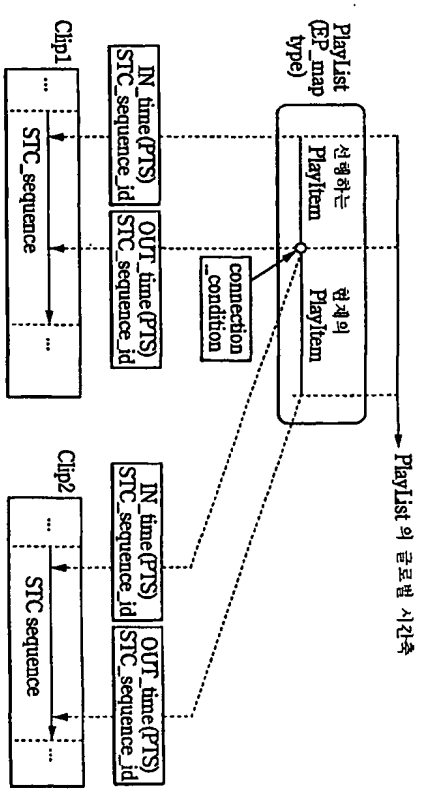
is_played_flag	의미
0b	그 PlayList는 기록되고 나서 한번도 재생된 적이 없다
1b	PlayList 는 기록되고 나서 한번은 재생되었다

B

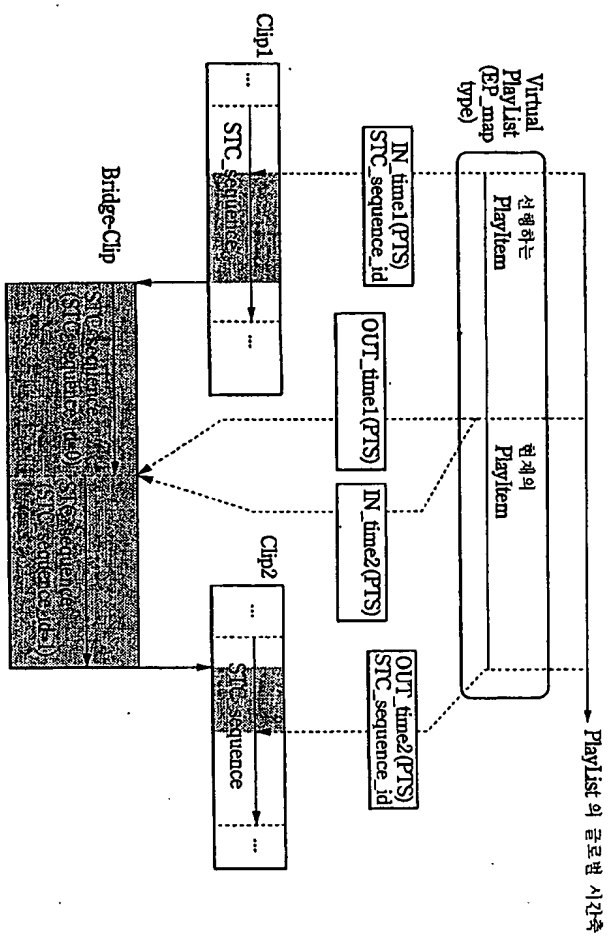
archive	의미
00b	어떠한 의미도 정의되어 있지 않다
01b	오리지널
10b	복사
11b	reserved

C

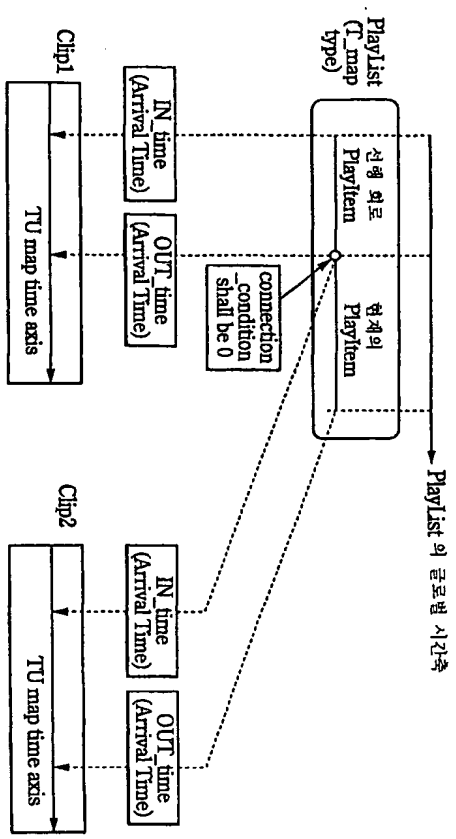
도면29


PlayList (EP_map type)
선행하는 PlayItem
현재의 PlayItem
connection_condition
PlayList 의 글로벌 시간축
Clip1
IN_time(PTS) STC_sequence_id
OUT_time(PTS) STC_sequence_id
STC_sequence
Clip2
IN_time(PTS) STC_sequence_id
OUT_time(PTS) STC_sequence_id
STC sequence

도면30


Virtual PlayList (EP_map type)
선행하는 PlayItem
현재의 PlayItem
IN_time1(PTS) STC_sequence_id
OUT_time1(PTS)
IN_time2(PTS)
OUT_time2(PTS) STC_sequence_id
PlayList 의 글로벌 시간축
Clip1
STC_sequence
Clip2
STC_sequence
Bridge-Clip

도면31



도면32

신택스	바이트수	약호
PlayItem(){		
Clip_information_file_name	8*10	bslbf
reserved	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
reserved	14	bslbf
connection_condition	2	bslbf
if (<Virtual PlayList>){		
if (*connection_condition=='10'*){		
BridgeSequenceInfo()		
}		
}		
}		

도면33

CPI_type in the PlayList()	IN_time의 시멘틱스
EP_map type	IN_time는、PlayItem 중에서 최후의 프레젠테이션 유닛에 대응하는 33비트 길이의 PTS의 상위 32비트를 나타내야 한다.
TU_map type	IN_time는、TU_map_time_axis 상의 시각이어야 한다. 또한 IN_time는、time_unit의 정밀도를 통합하여 나타내어야 한다. IN_time는 다음에 나타내는 등식에 의해 계산된다. $IN\_time = TU\_start\_time \%2^{32}$

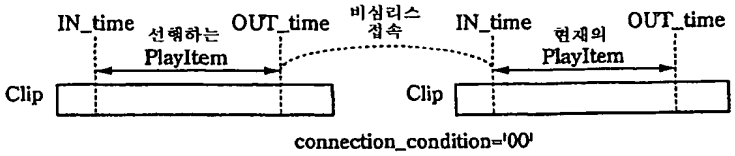
도면34

CPI_type in the PlayList()	OUT_time의 시멘틱스
EP_map type	OUT_time는 다음에 나타내는 등식에 의해 계산된다. Presentation_end_TS의 값의 상위 32비트를 나타내야 한다. $Presentation\_end\_TS = PTS\_out + AU\_duration$ 여기서, PTS_out는 PlayItem 중에서 최후의 프레젠테이션 유닛에 대응하는 33비트 길이의 PTS이다. AU_duration은 최후의 프레젠테이션 유닛의 90kHz 단위의 표시 기간이다.
TU_map type	OUT_time는 TU_map_time_axis 상의 시각이어야 한다. 또한 out_time는 time_unit의 정밀도로 통합해서 나타내어야 한다. out_time는 다음에 나타내는 등식에 의해 계산된다. $OUT\_time = TU\_start\_time \%2^{32}$

도면35

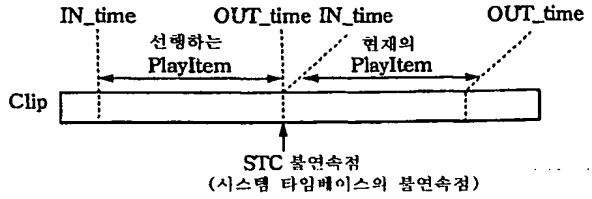
connection_condition	의미
00	· 선행하는 PlayItem과 현재의 PlayItem의 접속은 심리스 재생의 보증이 이루어져 있지 않다. · PlayList의 CPI_type가 TU_map type 인 경우, connection_condition 은 이 값을 리세트되어야 한다.
01	· 이 상태는 PlayList이 CPI_type가 EP_map type인 경우에만 허가된다. · 선행하는 PlayItem과 현재의 PlayItem은 시스템 타임 베이스 (STC베이스)의 불연속점이 있기 때문에 분할되어 있는 것을 나타낸다.
10	· 이 상태는 PlayList의 CPI_type가 EP_map type인 경우에만 허가된다. · 이 상태는 Virtual PlayList 에 대해서만 허가된다. · 선행하는 PlayItem과 현재의 PlayItem 과의 접속은 심리스 재생의 보증이 행해져 있다. · 선행하는 PlayItem 과 현재의 PlayItem 은 BridgeSequence 를 사용하여 접속되어 있고, DVR MPEG-2 트랜스포트 스트림은 후술하는 DVR-STD에 따라야 한다.
11	· 이 상태는 PlayList의 CPI_type가 EP_map type 인 경우에만 허가 된다 · 선행하는 PlayItem과 현재의 PlayItem 은 심리스 재생의 보증이 행해져 있다. · 선행하는 PlayItem 과 현재의 PlayItem은 BridgeSequence 를 사용하지 않고 접속되어 있고, DVR MPEG-2 트랜스포트 스트림은 후술하는 DVR-STD에 따라야 한다.

도면36



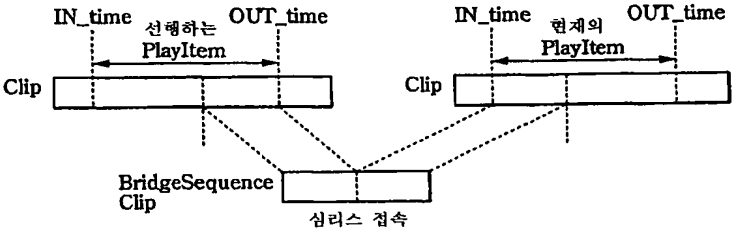
connection_condition='00'

A



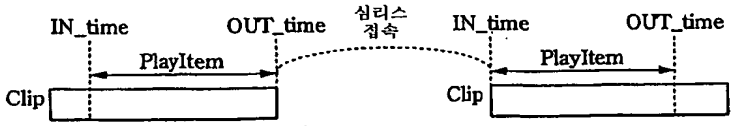
connection_condition='01'

B



connection_condition='10'

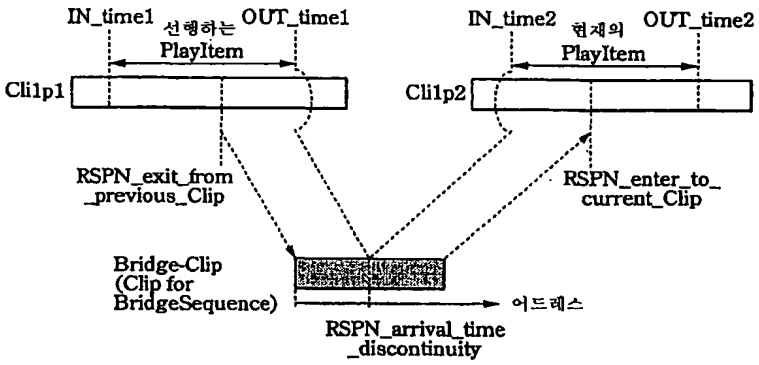
C



connection_condition='11'

D

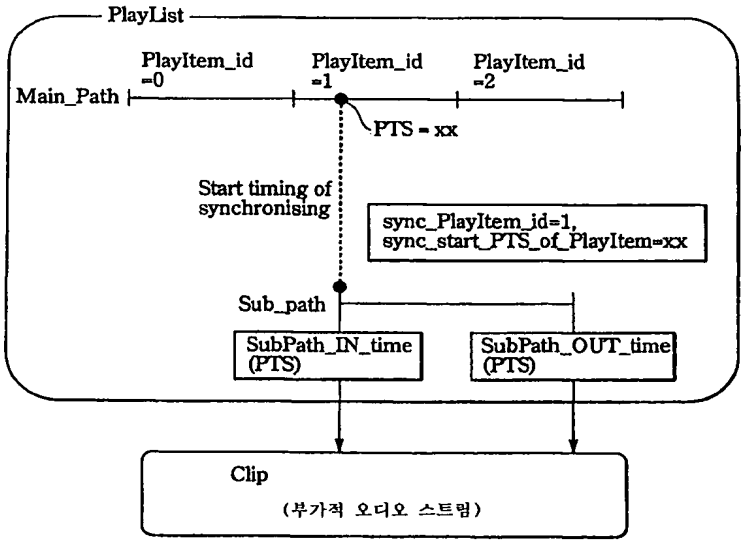
도면37



도면38

신택스	바이트수	약호
BridgeSequenceInfo() {		
Bridge_Clip_Information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

도면39



도면40

신택스	바이트수	약호
SubPlayItem(){		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

도면41

SubPath_type	의미
0x00	Auxiliary audio steam path
0x01-0xff	reserved

도면42

신택스	바이트수	약호
PlayListMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<*number_of_PlayList_marks*;i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_id	8	uimsbf
reserved	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

도면43

Mark_type	의미	코멘트
0x00	resume-mark	재생 리쥼 포인트. PlayListMark()에 있어서 정의 되는 재생 리쥼 포인트의 수는 0 또는 1이어야 한다.
0x01	book-mark	PlayList의 재생 엔트리 포인트. 이 마크는 사용자가 세트할 수 있고, 예를 들면 마음에 드는 장면의 개시점을 지정하는 마크를 사용한다.
0x02	skip-mark	스킵 마크 포인트. 이 포인트로부터 프로그램의 최후까지 플레이어는 프로그램을 스킵한다. PlayListMark()에 있어서 정의되는 스킵 마크 포인트의 수는 0 또는 1이어야 한다.
0x03-0x8F	reserved	
0x90-0xFF	reserved	Reserved for ClipMark()

도면44

CPI_type in the PlayList()	mark_time_stamp 의 시맨틱스
EP_map type	mark_time_stamp 는 마크에서 참조되는 프레젠테이션 유닛에 대응하는 33비트 길이의 PTS의 상위 32비트를 나타내야 한다.
TU_map type	mark_time_stamp 는 TU_map_time_axis 상의 시각이어야 한다. 또한 mark_time_stamp 는 time_unit 의 정밀도로 표현하여 나타내야 한다. mark_time_stamp 는 다음에 나타내는 등식에 의해 계산된다. $mark\_time\_stamp = TU\_start\_time \% 2^{32}$

도면45

신택스	바이트수	약호
zzzzz.clpi {		
STC_Info_Start_address	32	uimsbf
ProgramInfo_Start_address	32	uimsbf
CPI_Start_address	32	uimsbf
ClipMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	96	bslbf
ClipInfo()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
STC_Info()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for (i=0;i<N3;i++){		
padding_word	16	bslbf
}		
CPI()		
for (i=0;i<N4;i++){		
padding_word	16	bslbf
}		
ClipMark()		
for (i=0;i<N5;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

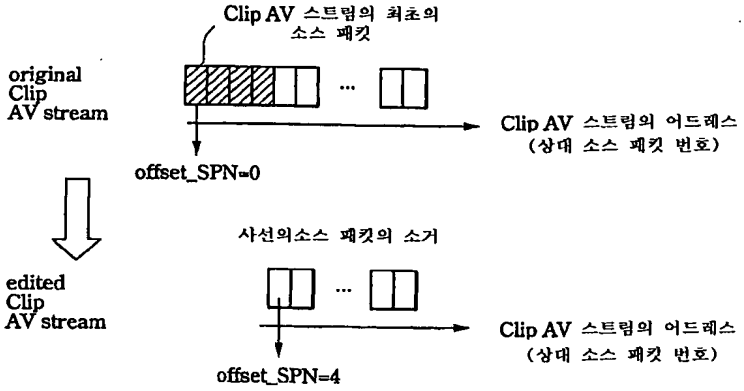
도면46

신택스	바이트수	약호
ClipInfo(){		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
if (Clip_stream_type==1) // Bridge-Clip AV stream		
RSPN_arrival_time_discontinuity	32	uimsbf
else		
reserved	32	bslbf
reserved_for_system_use	144	bslbf
reserved	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_service_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
service_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_fortune_use	256	bslbf
}		

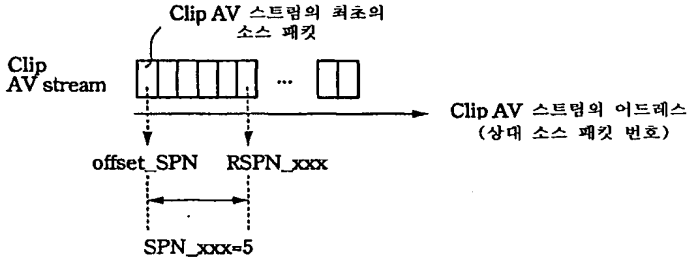
도면47

Clip_stream_type	의미
0	Clip AV 스트림
1	Bridge-Clip AV 스트림
2-255	Reserved

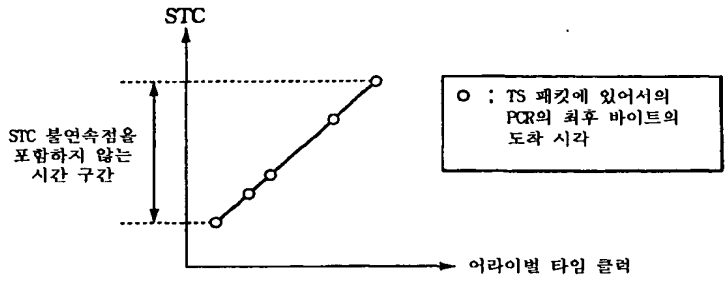
도면48


Clip AV 스트림의 최초의 소스 패킷
original Clip AV stream
Clip AV 스트림의 어드레스 (상대 소스 패킷 번호)
offset_SPN=0
사선의소스 패킷의 소거
edited Clip AV stream
Clip AV 스트림의 어드레스 (상대 소스 패킷 번호)
offset_SPN=4

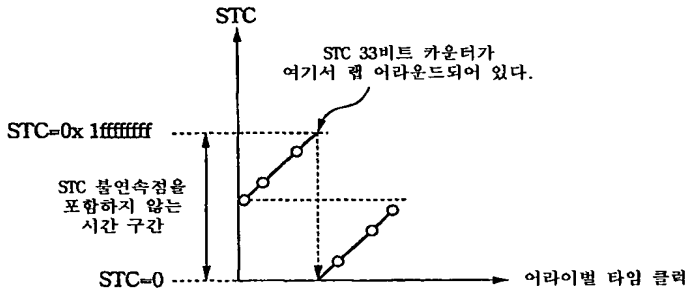
도면49


Clip AV 스트림의 최초의 소스 패킷
Clip AV stream
Clip AV 스트림의 어드레스 (상대 소스 패킷 번호)
offset_SPN
RSPN_xxx
SPN_xxx=5

도면50

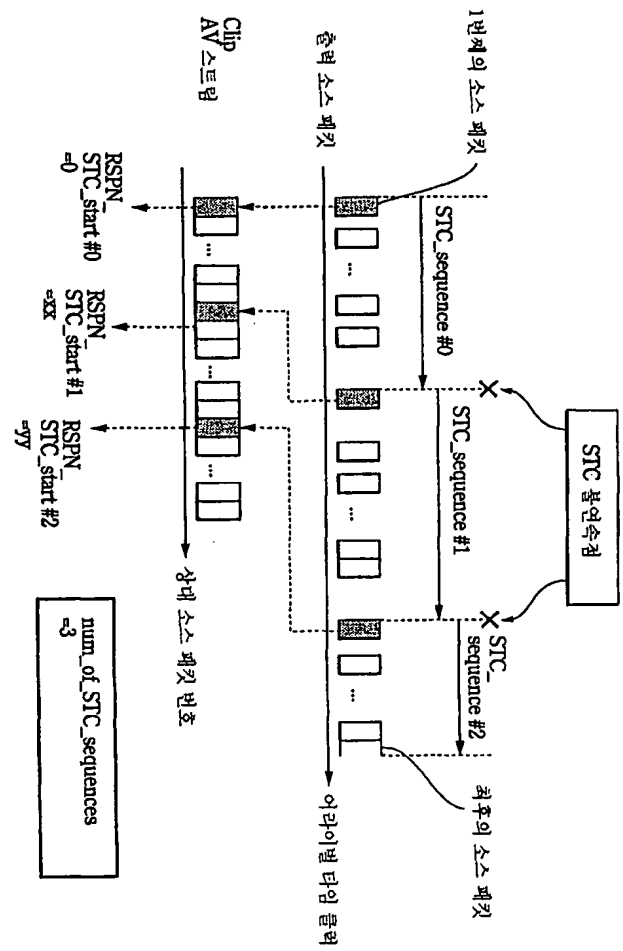

STC
STC 불연속점을
포함하지 않는
시간 구간
o : TS 패킷에 있어서의
PCR의 최후 바이트의
도착 시각
어라이벌 타임 클럭

A


STC
STC 33비트 카운터가
여기서 랩 어라운드되어 있다.
STC=0x 1ffffffff
STC 불연속점을
포함하지 않는
시간 구간
STC=0
어라이벌 타임 클럭

B

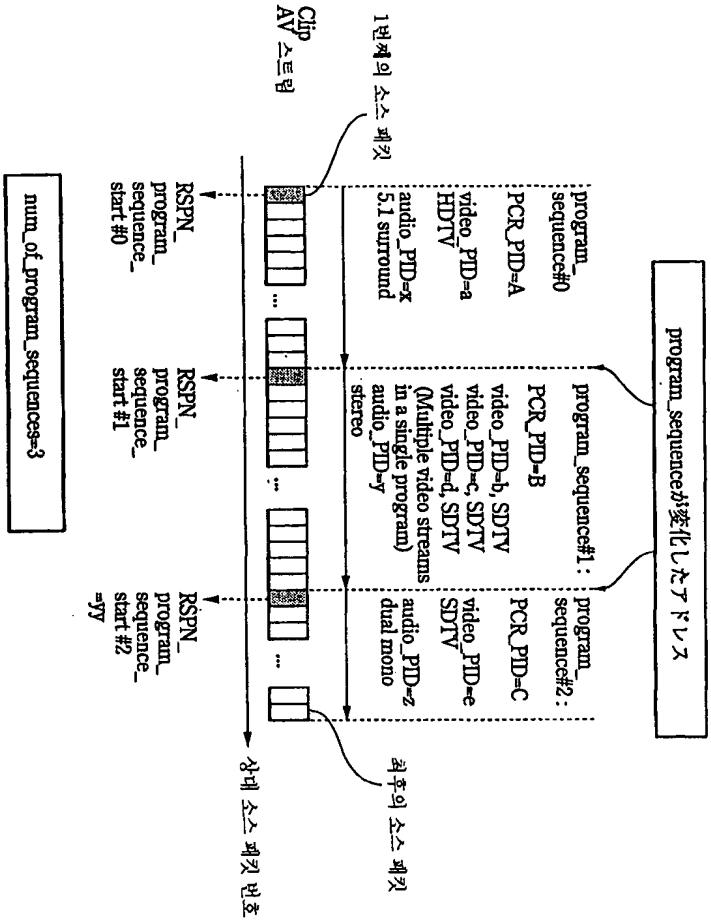
도면51



도면52

신택스	바이트수	약호
STC_Info(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
num_of_STC_sequences	8	uimsbf
for (*STC_sequence_id=0;* *STC_sequence_id<num_of_STC_sequences;* *STC_sequence_id++*){		
resereved	32	bslbf
RSPN_STC_start	32	uimsbf
}		
}		
}		

도면53


num_of_program_sequences=3
RSPN_program_sequence_start #0
RSPN_program_sequence_start #1
RSPN_program_sequence_start #2 =yy
Clip AV 스트림
1번째의 소스 패킷
program_sequence#0
PCR_PID=A
video_PID=a HDTV
audio_PID=x 5.1 surround
program_sequence#1 :
PCR_PID=B
video_PID=b, SDTV
video_PID=c, SDTV
video_PID=d, SDTV (Multiple video streams in a single program)
audio_PID=y stereo
program_sequence#2 :
PCR_PID=C
video_PID=e SDTV
audio_PID=z dual mono
program_sequence가 変化したアドレス
상대 소스 패킷 번호
최후의 소스 패킷

도면54

신택스	바이트수	약호
ProgramInfo(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
number_of_program_sequences	8	uimsbf
for (i=0;i<*number_of_program_sequences*;i++){		
RSPN_program_sequence_start	32	uimsbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0;k<*number_of_videos*;k++){		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0;k<*number_of_audios*;k++){		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		
}		

도면55

신택스	바이트수	약호
VideoCodingInfo() {		
video_format	8	uimsbf
frame_rate	8	uimsbf
display_aspect_ratio	8	uimsbf
reserved	8	bslbf
}		

도면56

video_format	의미
0	480i
1	576i
2	480p(including 640×480p format)
3	1080i
4	720p
5	1080p
6-254	reserved
255	No information

도면57

frame_rate	의미
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94..)
8	60
9-254	reserved
255	No information

도면58

display_aspect_ratio	의미
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ration
4-254	reserved
255	No information

도면59

신택스	바이트수	약호
AudioCodingInfo() {		
audio_format	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
reserved	8	bslbf
}		

도면60

audio_coding	의미
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

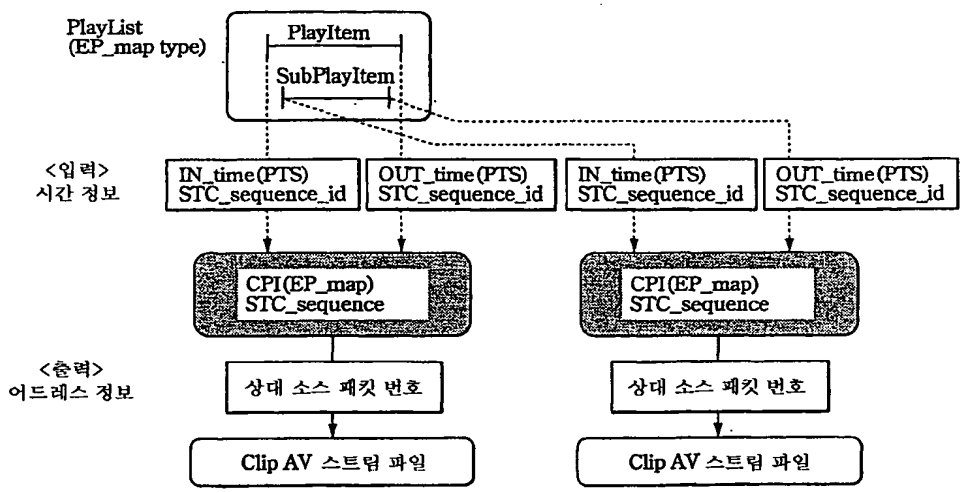
도면61

audio_component_type	의미
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

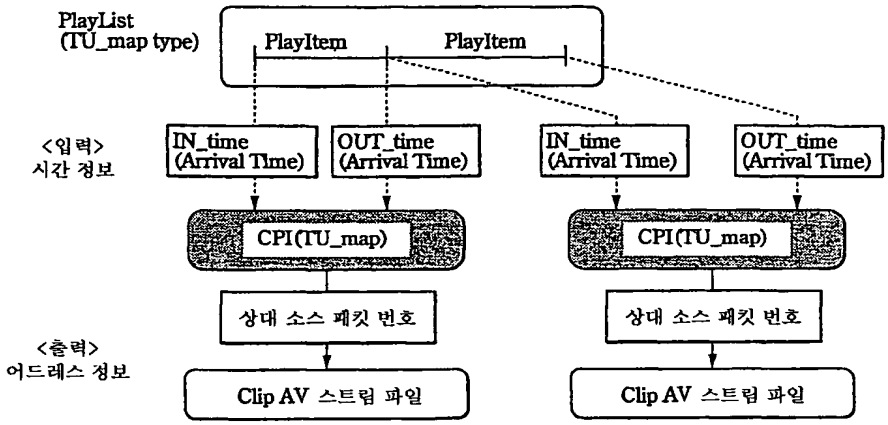
도면62

sampling_frequency	의미
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

도면63



도면64



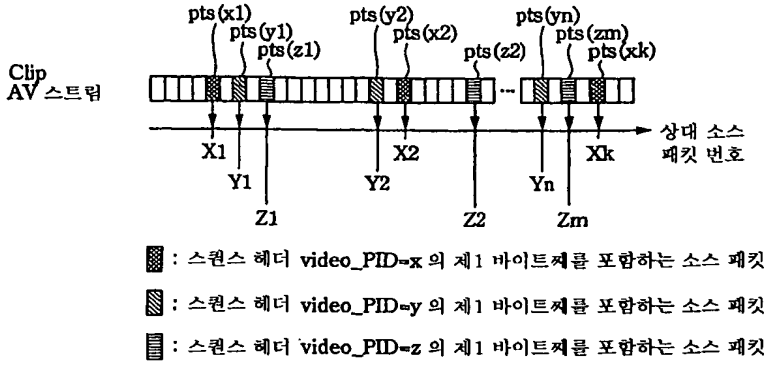
도면65

신택스	바이트수	약호
CPI(){		
version_number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI_type	1	bslbf
if (CPI_type==0)		
EP_map()		
else		
TU_map()		
}		

도면66

CPI_type	의미
0	EP map type
1	TU map type

도면67



EP_map
number_of_stream/PIDs=3

EP_map_for_one_stream_PID(0)
stream_PID(0)=x
num_EP_entries(0)=k

PTS_EP start	RSPN_EP start
pts(x1)	X1
pts(x2)	X2
...	...
pts(xk)	Xk

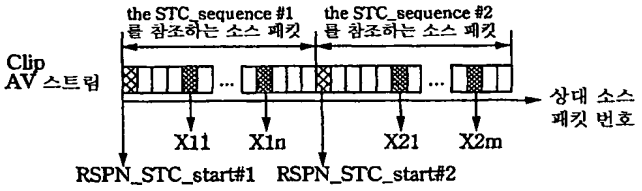
EP_map_for_one_stream_PID(1)
stream_PID(1)=y
num_EP_entries(1)=n

PTS_EP start	RSPN_EP start
pts(y1)	Y1
pts(y2)	Y2
...	...
pts(yn)	Yn

EP_map_for_one_stream_PID(2)
stream_PID(2)=z
num_EP_entries(2)=m

PTS_EP start	RSPN_EP start
pts(z1)	Z1
pts(z2)	Z2
...	...
pts(zm)	Zm

도면68



: 스퀀스 헤더 video_PID=x 의 제1 바이트째를 포함하는 소스 패킷

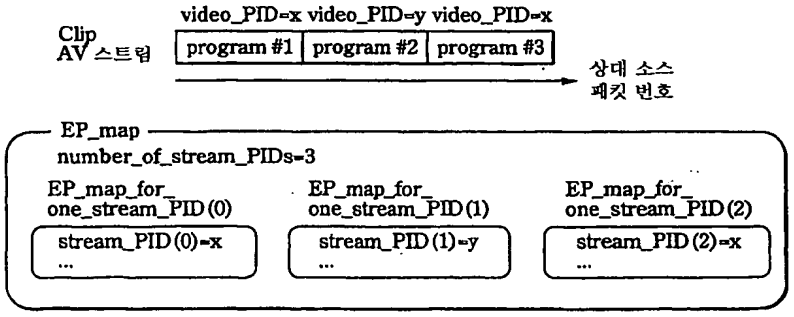
: RSPN_STC_start 에 의해 참조되는 소스 패킷
(the STC_into 에 있어서 정의된다)

EP_map_for_one_steram_ PID
video_PID=x

PTS_EP start	RSPN_EP start	
pts(x11)	X11	STC_sequence #1 에 속하는 데이터
...	...	
pts(x1n)	X1n	
		→ boundary
pts(x21)	X21	STC_sequence #2 에 속하는 데이터
...	...	
pts(x2m)	X2m	

RSPN_STC_start #2<X21

도면69



도면70

신택스	바이트수	약호
EP_map(){		
reserved	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<*number_of_stream_PIDs*;k++){		
stream_PID(*k*)	16	bslbf
num_EP_entries(*k*)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(*k*)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k<*number_of_stream_PIDs*;k++){		
EP_map_for_one_stream_PID(*num_EP_entries(k)*)		
for (i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

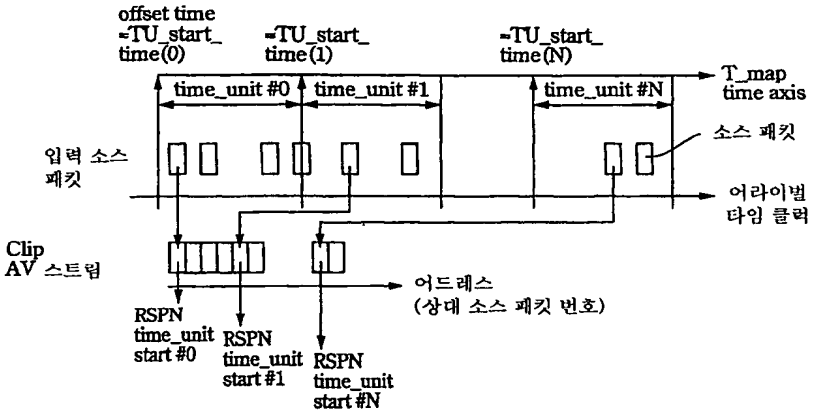
도면71

EP_type	의미
0	video
1	audio
2-15	reserved

도면72

신택스	바이트수	약호
EP_map_for_one_stream_PID(N) {		
for (i=0;i<N;i++) {		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

도면73



도면74

신택스	바이트수	약호
TU_map() {		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<*number_of_time_unit_entries*;k++)		
RSPN_time_unit_start	32	uimsbf
}		

도면75

신택스	바이트수	약호
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<*number_of_clip_marks*; i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

도면76

Mark_type	의미	코멘트
0x00-0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	프로그램의 개시 포인트를 나타내는 마크점
0x91	Local event-start mark	프로그램 중의 국소적인 장면을 나타내는 마크점
0x92	Scene-start mark	장면 전환 포인트를 나타내는 마크점
0x93-0xFF	reserved	

도면77

CPI_type in the CPI()	mark_time_stamp 의 시맨틱스
EP_map type	mark_time_stamp 는 마크로 참조되는 프레젠테이션 유닛에 대응하는 33비트 길이의 PTS의 상위 32비트를 나타내야만 한다
TU_map type	mark_time_stamp 는 TU_map_time_axis 상의 시각이어야만 한다. 또한 mark_time_stamp 는 time_unit 의 정밀도로 통합하여 나타내어야만 한다. mark_time_stamp 는 다음에 나타내는 등식에 의해 계산된다 $mark\_time\_stamp = TU\_start\_time \% 2^{32}$

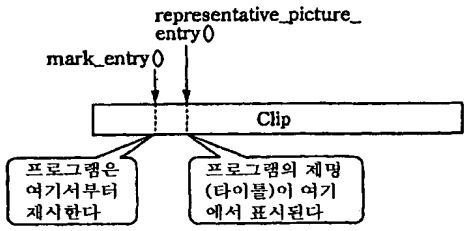
도면78

신택스	바이트수	약호
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<*number_of_Clip_marks*; i++){		
reserved	8	bslbf
mark_type	8	bslbf
reserved_for_MakerID	16	bslbf
mark_entry()		
representative_picture_entry()		
ref_thumbnail_index	16	uimsbf
}		
}		

도면79

Mark_type	의미	코멘트
0x00-0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	프로그램의 개시 포인트를 나타내는 마크점
0x91	Local event-start mark	프로그램 중의 국소적인 장면을 나타내는 마크점
0x92	Scene-start mark	장면 개시 포인트를 나타내는 마크점
0x93	Scene-end mark	장면 종료 포인트를 나타내는 마크점
0x94	CM-start mark	CM 개시 포인트를 나타내는 마크점
0x95	CM-end mark	CM 종료 포인트를 나타내는 마크점
0x96-0xBF	DVR 포맷이 Clip Mark를 장래, 확장할 때를 위해 예약되어 있다	
0xC0-0xFF	메이커 독자의 어플레케이션으로 이용하는 마크에 할당 가능	

도면80



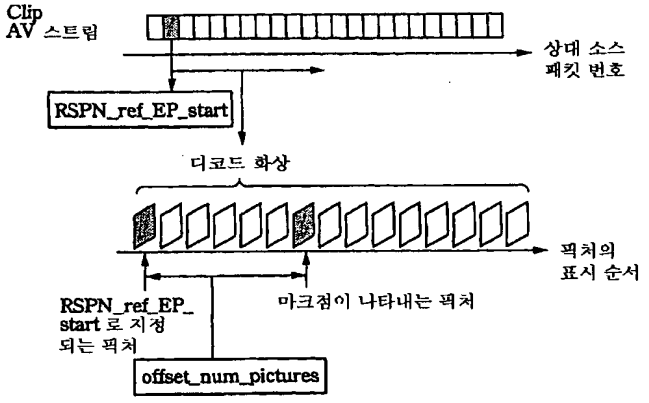
도면81

신택스	바이트수	약호
mark_entry()/representative_picture_entry(){		
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
}		

도면82

신택스	바이트수	약호
mark_entry()/representative_picture_entry(){		
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
}		

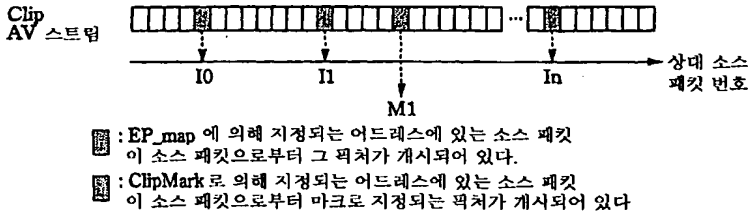
도면83



도면84

신택스	바이트수	약호
mark_entry()/representative_picture_entry(){		
RSPN_mark_point	32	uimsbf
}		

도면85



도면86

신택스	바이트수	약호
menu.thmb/mark.thmb() {		
reserved	256	bslbf
Thumbnail()		
for (i=0;i<N1;i++)		
padding_word	16	bslbf
}		

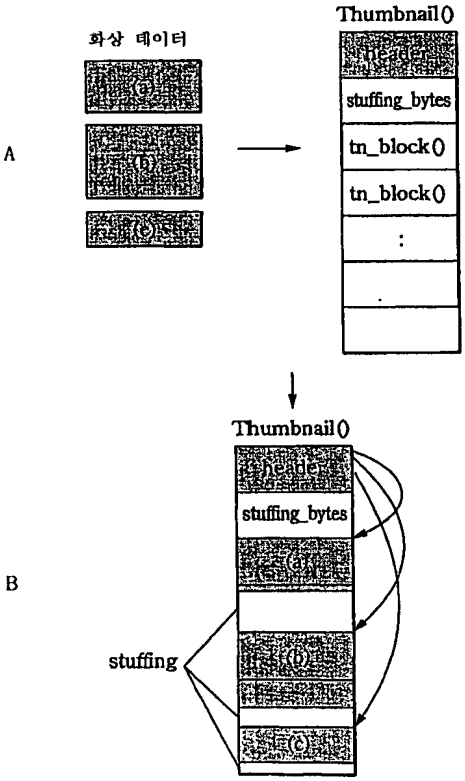
도면87

신택스	바이트수	약호
Thumbnail(){		
version_number	8*4	char
length	32	uimsbf
if (length !=0){		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<*number_of_thumbnails*; i++){		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k=0; k<*number_of_tn_blocks*; k++){		
tn_block	tn_block_ size*1024*8	
}		
}		
}		

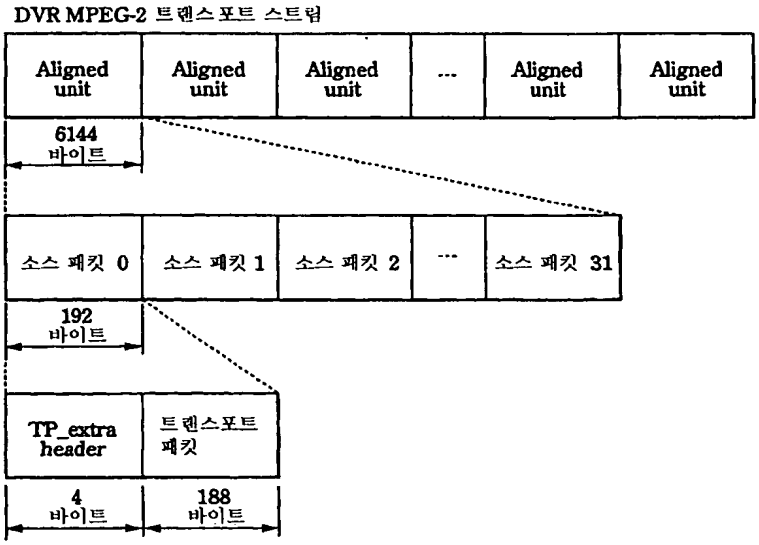
도면88

Thumbnail_picture_format	의미
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

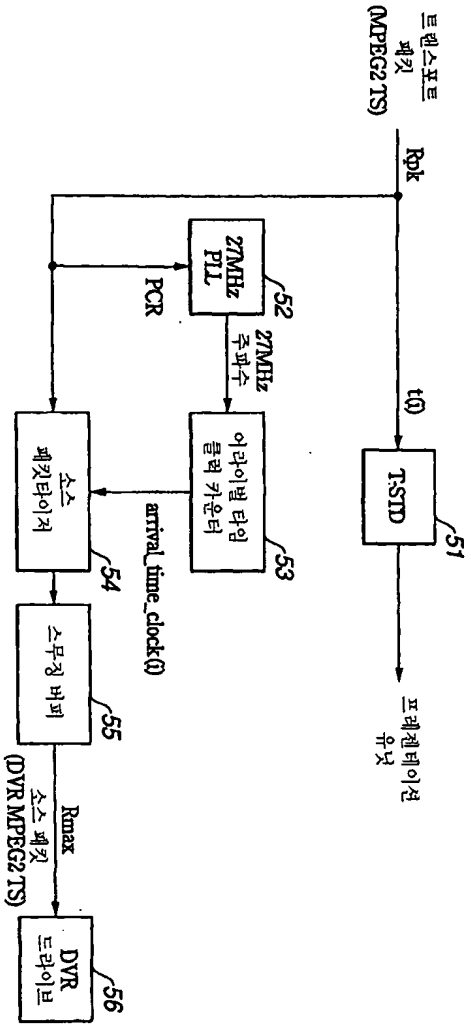
도면89



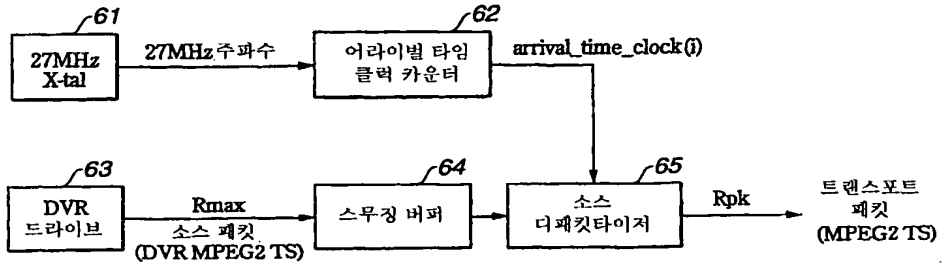
도면90


DVR MPEG-2 트랜스포트 스트림
Aligned unit
Aligned unit
Aligned unit
...
Aligned unit
Aligned unit
6144 바이트
소스 패킷 0
소스 패킷 1
소스 패킷 2
...
소스 패킷 31
192 바이트
TP_extra header
트랜스포트 패킷
4 바이트
188 바이트

도면91


트랜스포트 패킷 (MPEG2 TS)
Rpk
27MHz PLL
52
PCR
27MHz 주파수
t(i)
T-STD
51
소스 패킷타이저
54
arrival_time_clock(i)
어라이벌 타임 클럭 카운터
53
스무딩 버퍼
55
프레젠테이션 유닛
Rmax
소스 패킷 (DVR MPEG2 TS)
DVR 드라이브
56

도면92


61
27MHz X-tal
27MHz주파수
62
어라이벌 타임 클럭 카운터
arrival_time_clock(i)
63
DVR 드라이브
Rmax
소스 패킷 (DVR MPEG2 TS)
64
스무딩 버퍼
65
소스 디패킷타이저
Rpk
트랜스포트 패킷 (MPEG2 TS)

도면93

신택스	바이트수	약호
source_packet() {		
TP_extra_header()		
trasport_packet()		
}		

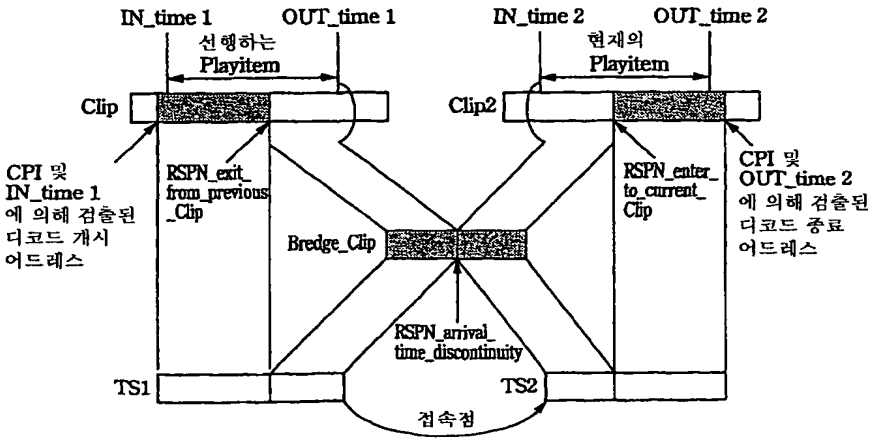
도면94

신택스	바이트수	약호
TP_extra_header() {		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

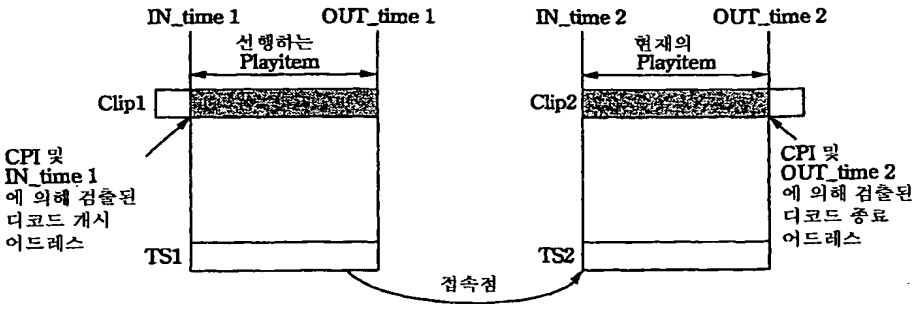
도면95

copy_permission _indicator	의미
00	copy free
01	no more copy
10	copy once
11	copy prohibited

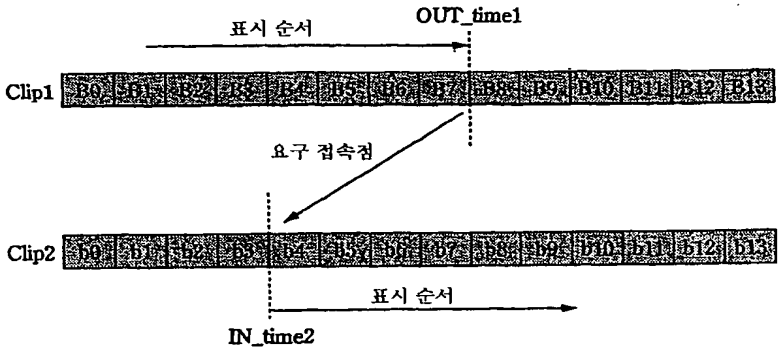
도면96



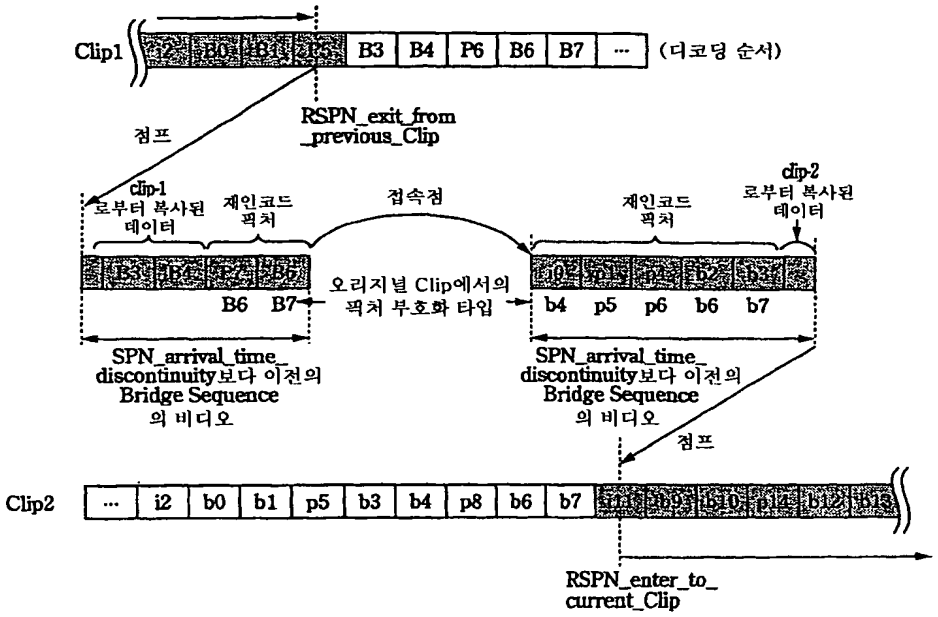
도면97


IN_time 1
OUT_time 1
IN_time 2
OUT_time 2
선행하는
Playitem
현재의
Playitem
Clip1
Clip2
CPI 및
IN_time 1
에 의해 검출된
디코드 개시
어드레스
CPI 및
OUT_time 2
에 의해 검출된
디코드 종료
어드레스
TS1
TS2
접속점

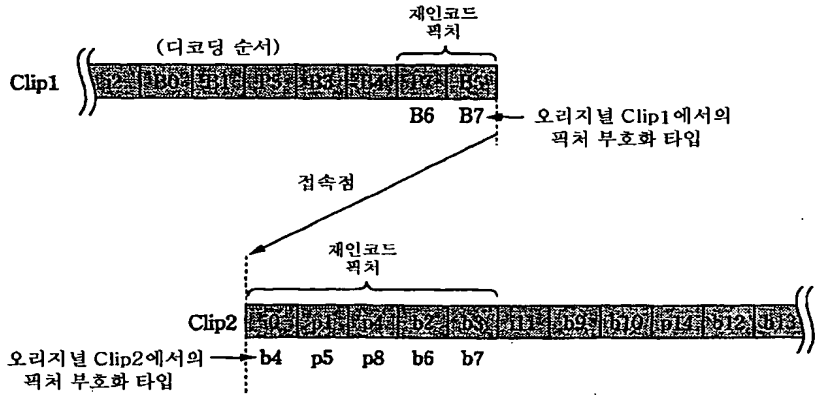
도면98


표시 순서
OUT_time1
Clip1
b0
b1
b2
b3
b4
b5
b6
b7
b8
b9
b10
b11
b12
b13
요구 접속점
Clip2
b0
b1
b2
b3
b4
b5
b6
b7
b8
b9
b10
b11
b12
b13
표시 순서
IN_time2

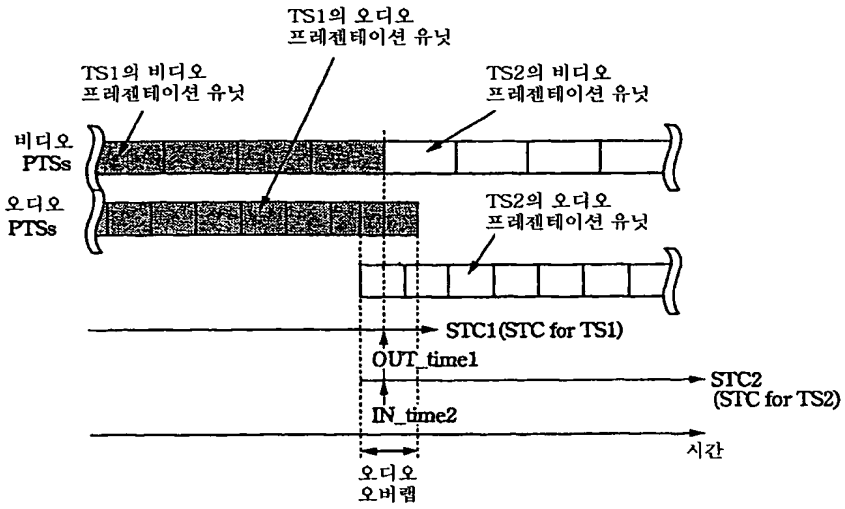
도면99


Clip1
B3
B4
P6
B6
B7
...
(디코딩 순서)
점프
RSPN_exit_from
_previous_Clip
clip-1
로부터 복사된
데이터
재인코드
픽처
접속점
재인코드
픽처
clip-2
로부터 복사된
데이터
B6
B7
오리지널 Clip에서의
픽처 부호화 타입
b4
p5
p6
b6
b7
SPN_arrival_time_
discontinuity보다 이전의
Bridge Sequence
의 비디오
SPN_arrival_time_
discontinuity보다 이전의
Bridge Sequence
의 비디오
점프
Clip2
...
i2
b0
b1
p5
b3
b4
p8
b6
b7
RSPN_enter_to_
current_Clip

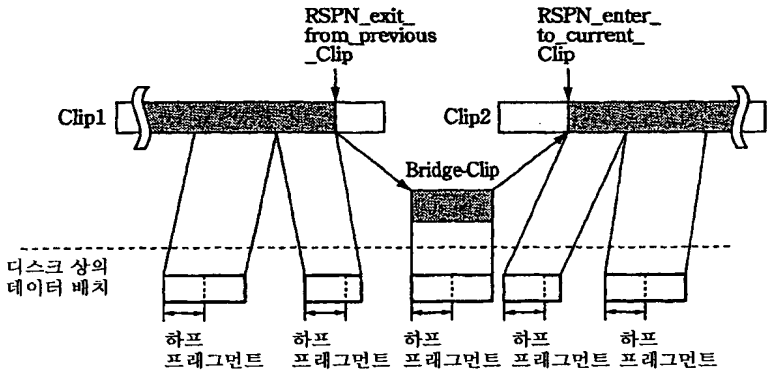
도면100


재인코드
픽처
(디코딩 순서)
Clip1
B6
B7
오리지널 Clip1에서의
픽처 부호화 타입
접속점
재인코드
픽처
Clip2
오리지널 Clip2에서의
픽처 부호화 타입
b4
p5
p8
b6
b7

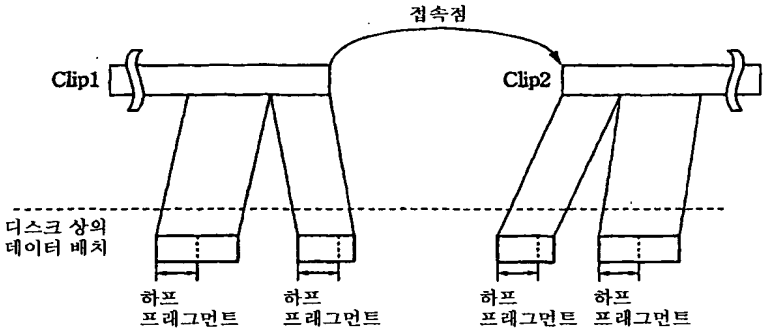
도면101



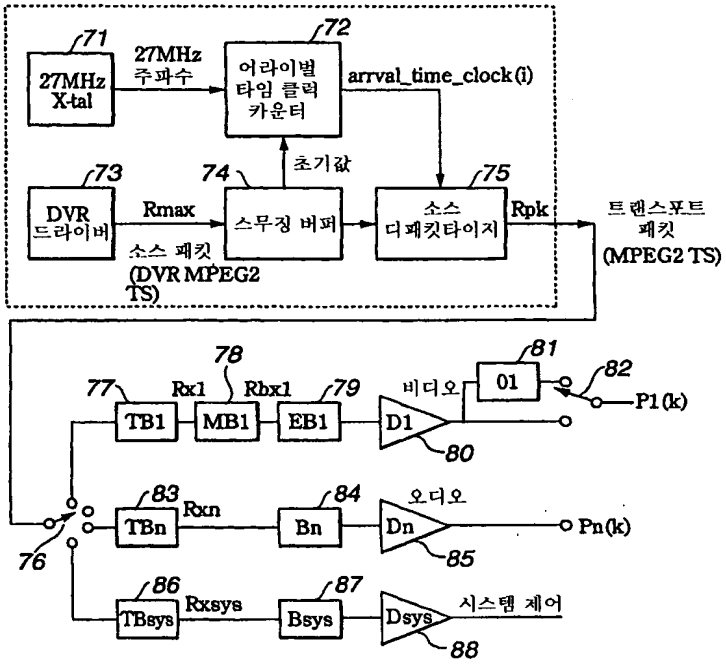
도면102



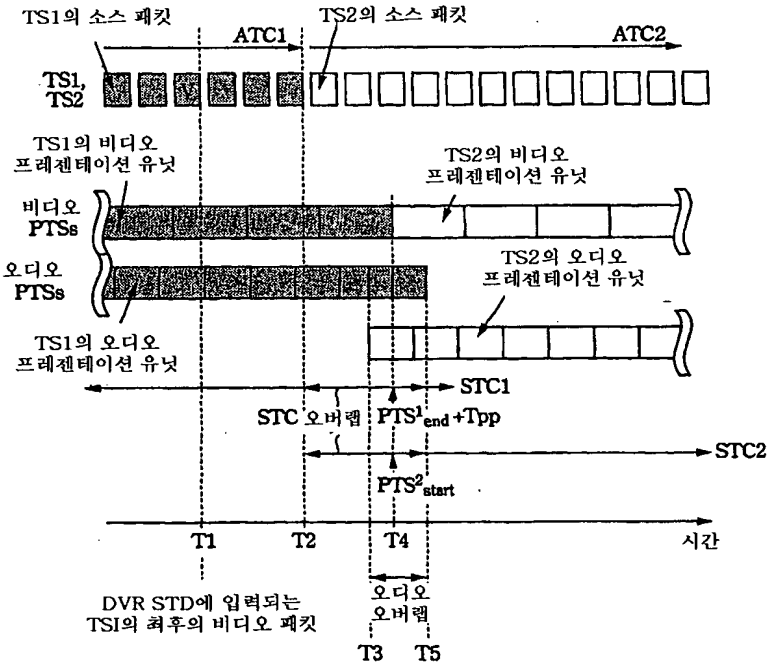
도면103



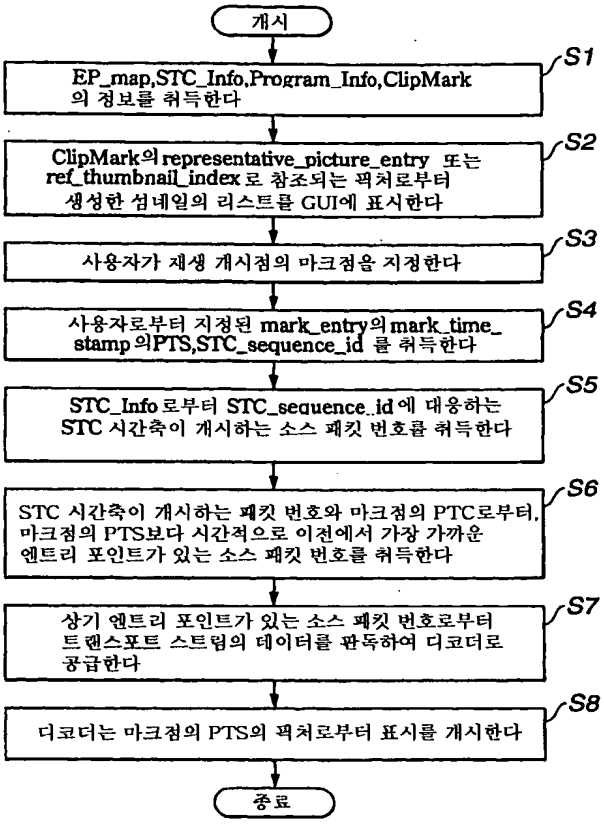
도면104


71
27MHz X-tal
27MHz 주파수
72
어라이벌 타임 클럭 카운터
arrval_time_clock(i)
초기값
73
DVR 드라이버
74
Rmax
스무징 버퍼
75
소스 디패킷타이지
Rpk
트랜스포트 패킷 (MPEG2 TS)
소스 패킷 (DVR MPEG2 TS)
78
77
Rx1
Rbx1
79
TB1
MB1
EB1
D1
80
비디오
81
01
82
P1(k)
83
TBn
Rxn
84
Bn
Dn
85
오디오
Pn(k)
76
86
TBsys
Rxsys
87
Bsys
Dsys
88
시스템 제어

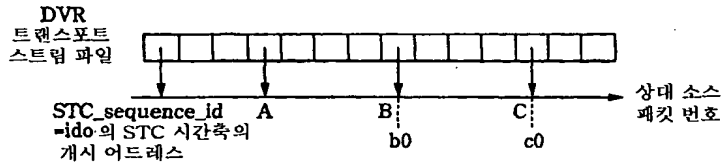
도면105


TS1의 소스 패킷
ATC1
TS2의 소스 패킷
ATC2
TS1, TS2
TS1의 비디오 프레젠테이션 유닛
TS2의 비디오 프레젠테이션 유닛
비디오 PTSs
오디오 PTSs
TS2의 오디오 프레젠테이션 유닛
TS1의 오디오 프레젠테이션 유닛
STC1
STC 오버랩
PTS1end+Tpp
STC2
PTS2start
T1
T2
T4
시간
DVR STD에 입력되는 TSI의 최후의 비디오 패킷
오디오 오버랩
T3
T5

도면106



도면107



도면108

EP_map

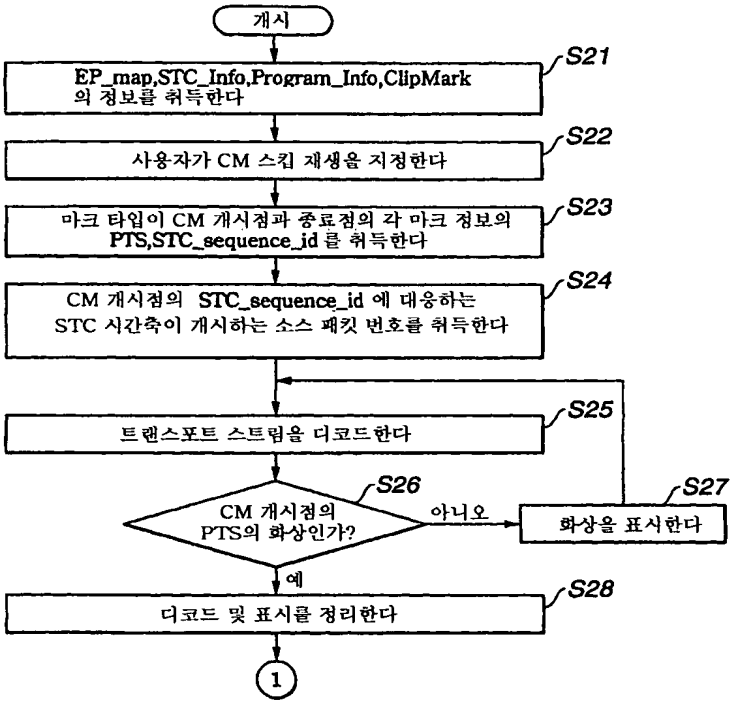
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

도면109

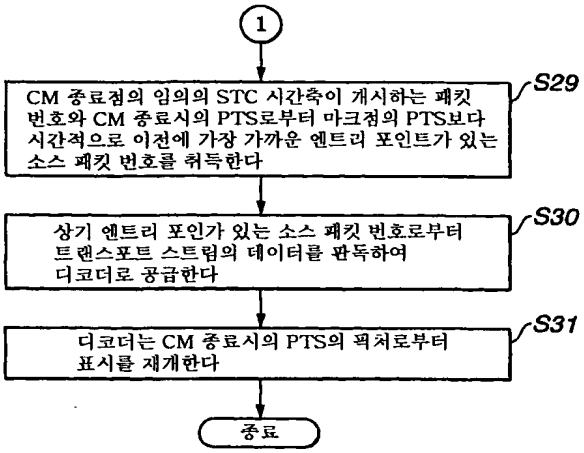
ClipMark

Mark_type	mark_entry		representative _ picture _entry	
	Mark_ Time_stamp	STC_sequence_ id	Mark_ Time_stamp	STC_sequence_ id
...	...	...		
0x92(scene start)	PTS(a1)	id0	PTS(a2)	id0
0x94(CM start)	PTS(b0)	id0	PTS(b0)	id0
0x95(CM end)	PTS(c0)	id0	PTS(c0)	id0
...	...	...	...	...

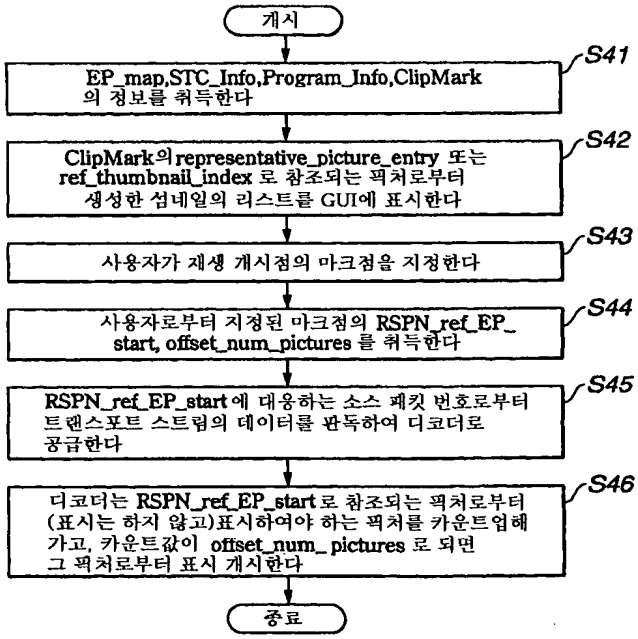
도면110



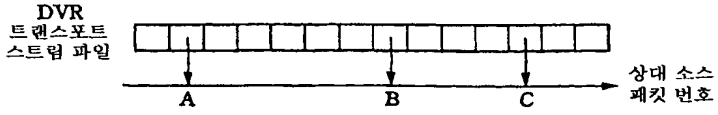
도면111



도면112



도면113



도면114

EP_map

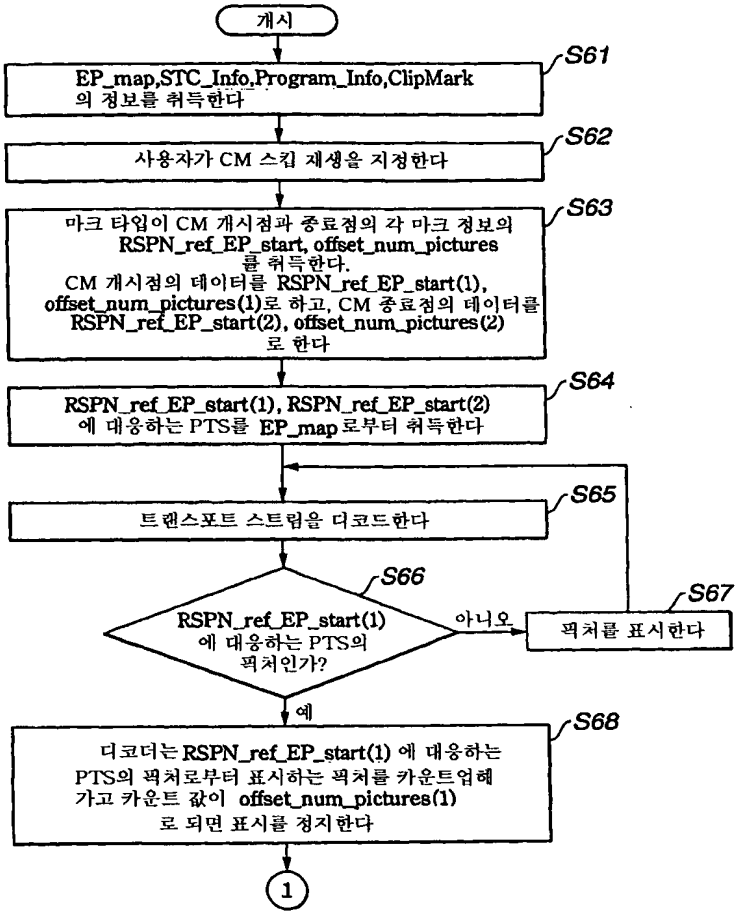
RSPN_EP_ start	PTS_EP_ start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

도면115

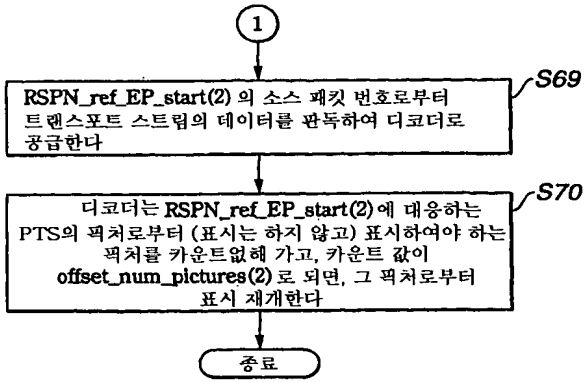
ClipMark

mark_type	mark_entry		representative _ picture _entry	
	RSPN_ref_EP_ start	offset_num_ pictures	RSPN_ref_EP_ start	offset_num_ pictures
...	...	...	...	...
0x92(scene start)	A	M1	A	M2
0x94(CM start)	B	N1	B	N1
0x95(CM end)	C	N2	C	N2
...	...	...	...	...

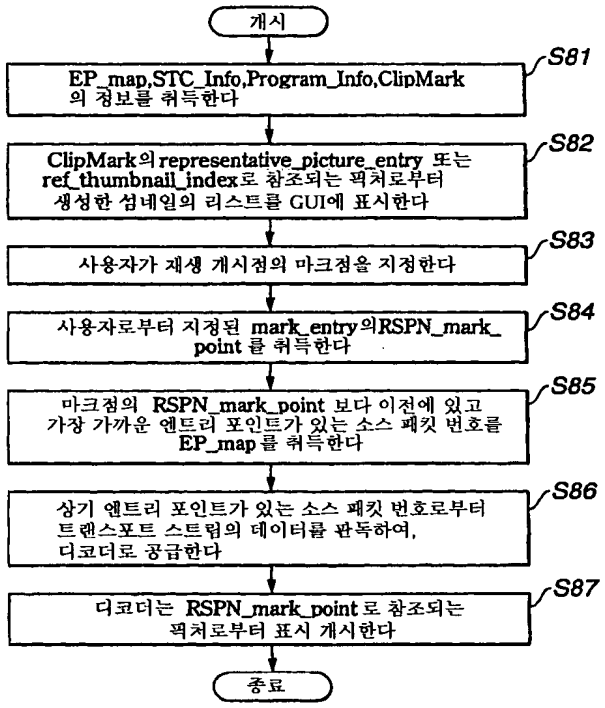
도면116


개시
EP_map,STC_Info,Program_Info,ClipMark 의 정보를 취득한다
S61
사용자가 CM 스킵 재생을 지정한다
S62
마크 타입이 CM 개시점과 종료점의 각 마크 정보의 RSPN_ref_EP_start, offset_num_pictures 를 취득한다. CM 개시점의 데이터를 RSPN_ref_EP_start(1), offset_num_pictures(1)로 하고, CM 종료점의 데이터를 RSPN_ref_EP_start(2), offset_num_pictures(2) 로 한다
S63
RSPN_ref_EP_start(1), RSPN_ref_EP_start(2) 에 대응하는 PTS를 EP_map 로부터 취득한다
S64
트랜스포트 스트림을 디코드한다
S65
RSPN_ref_EP_start(1) 에 대응하는 PTS의 픽처인가?
S66
아니오
픽처를 표시한다
S67
예
디코더는 RSPN_ref_EP_start(1) 에 대응하는 PTS의 픽처로부터 표시하는 픽처를 카운트업해 가고 카운트 값이 offset_num_pictures(1) 로 되면 표시를 정지한다
S68
1

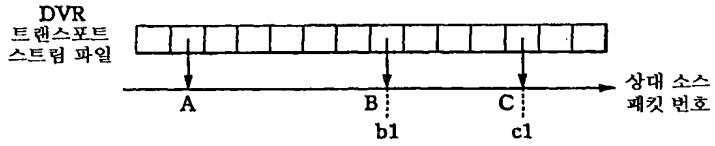
도면117


1
RSPN_ref_EP_start(2) 의 소스 패킷 번호로부터 트랜스포트 스트림의 데이터를 판독하여 디코더로 공급한다
S69
디코더는 RSPN_ref_EP_start(2) 에 대응하는 PTS의 픽처로부터 (표시는 하지 않고) 표시하여야 하는 픽처를 카운트업해 가고, 카운트 값이 offset_num_pictures(2) 로 되면, 그 픽처로부터 표시 재개한다
S70
종료

도면118



도면119



도면120

EP_map

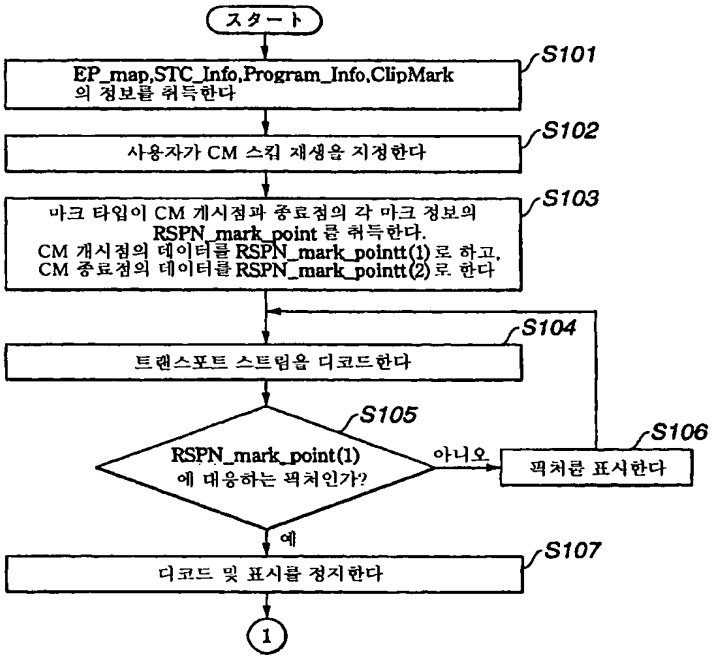
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

도면121

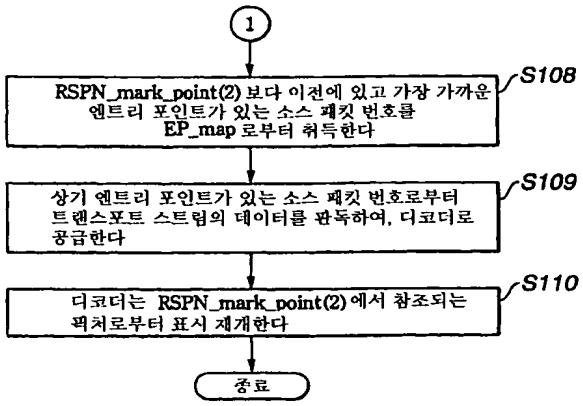
ClipMark

mark_type	mark_entry	representative _ picture _entry
	RSPN_mark_point	RSPN_mark_point
... 0x92(scene start) 0x94(CM start) 0x95(CM end) ...	... a1 b1 c1 ...	... a2 b1 c1 ...

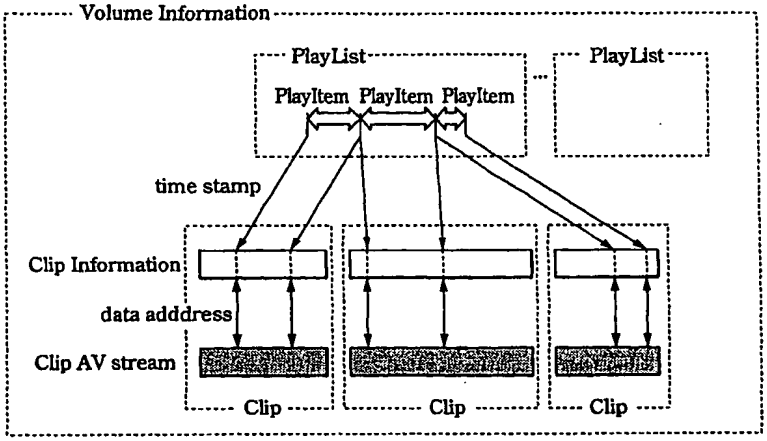
도면122



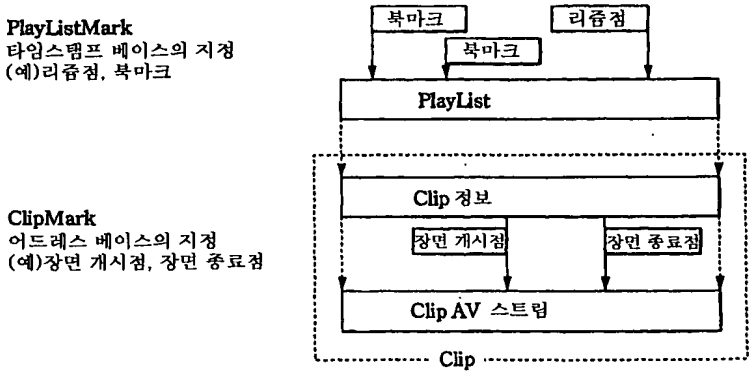
도면123



도면124



도면125



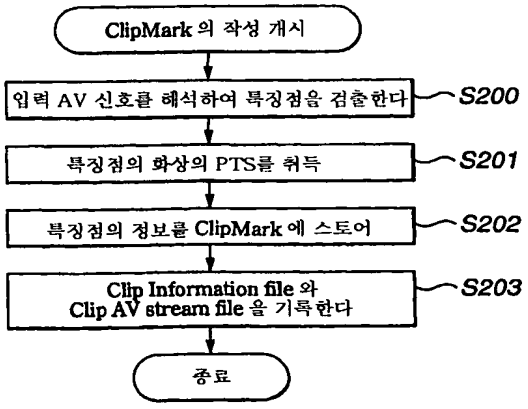
도면126

신택스	바이트수	약호
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<*number_of_Clip_marks*; i++){		
reserved	8	bslbf
mark_type	8	bslbf
RSPN_mark	32	uimsbf
reserved	32	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

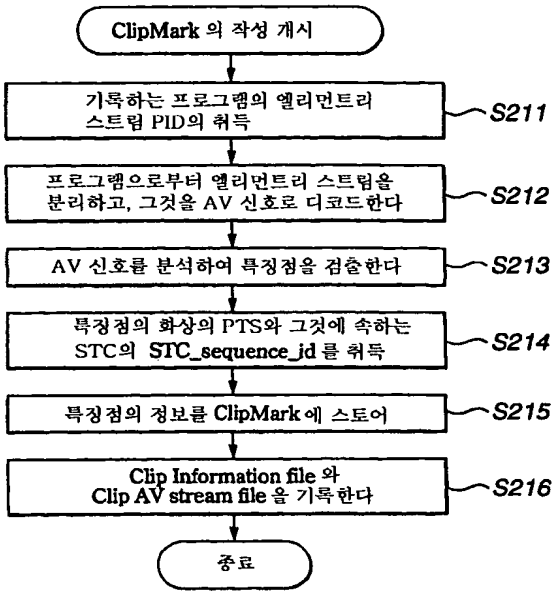
도면127

신택스	바이트수	약호
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<*number_of_Clip_marks*; i++){		
reserved	8	bslbf
mark_type	8	bslbf
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
ref_thumbnail_index	16	uimsbf
}		
}		

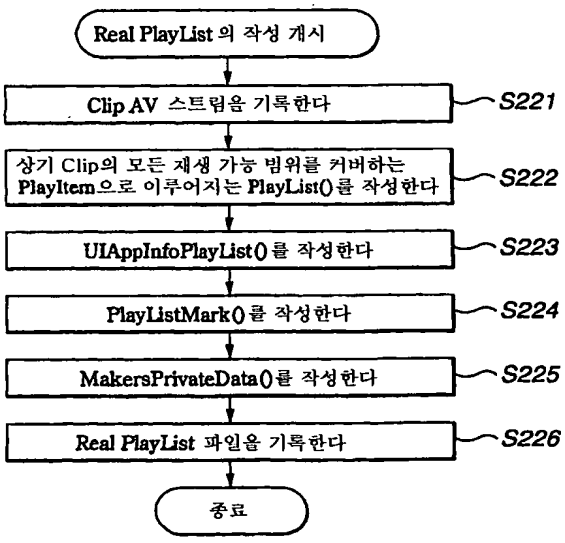
도면128



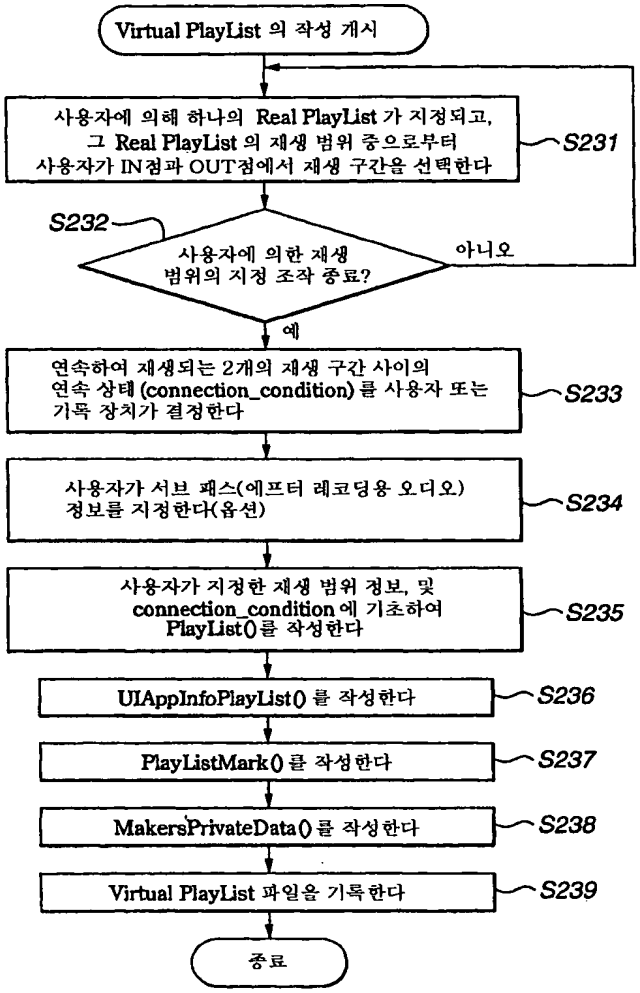
도면129


ClipMark 의 작성 개시
기록하는 프로그램의 엘리먼트리 스트림 PID의 취득
S211
프로그램으로부터 엘리먼트리 스트림을 분리하고, 그것을 AV 신호로 디코드한다
S212
AV 신호를 분석하여 특징점을 검출한다
S213
특징점의 화상의 PTS와 그것에 속하는 STC의 STC_sequence_id 를 취득
S214
특징점의 정보를 ClipMark 에 스토어
S215
Clip Information file 와 Clip AV stream file 을 기록한다
S216
종료

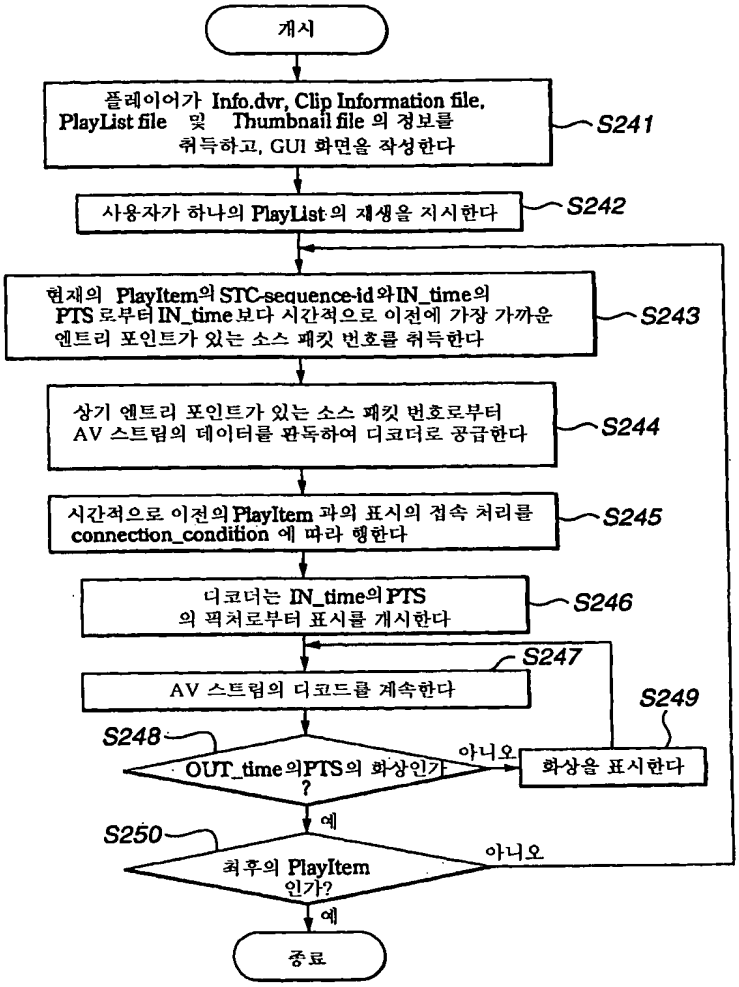
도면130


Real PlayList 의 작성 개시
Clip AV 스트림을 기록한다
S221
상기 Clip의 모든 재생 가능 범위를 커버하는 PlayItem으로 이루어지는 PlayList()를 작성한다
S222
UIAppInfoPlayList()를 작성한다
S223
PlayListMark()를 작성한다
S224
MakersPrivateData()를 작성한다
S225
Real PlayList 파일을 기록한다
S226
종료

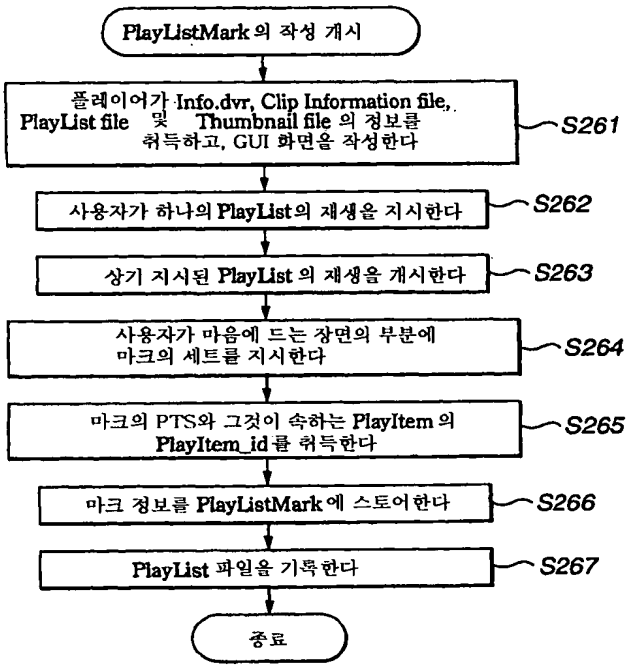
도면131


Virtual PlayList 의 작성 개시
사용자에 의해 하나의 Real PlayList 가 지정되고, 그 Real PlayList 의 재생 범위 중으로부터 사용자가 IN점과 OUT점에서 재생 구간을 선택한다
S231
S232
사용자에 의한 재생 범위의 지정 조작 종료?
아니오
예
연속하여 재생되는 2개의 재생 구간 사이의 연속 상태 (connection_condition) 를 사용자 또는 기록 장치가 결정한다
S233
사용자가 서브 패스(에프터 레코딩용 오디오) 정보를 지정한다(옵션)
S234
사용자가 지정한 재생 범위 정보, 및 connection_condition 에 기초하여 PlayList() 를 작성한다
S235
UIAppInfoPlayList() 를 작성한다
S236
PlayListMark() 를 작성한다
S237
MakersPrivateData() 를 작성한다
S238
Virtual PlayList 파일을 기록한다
S239
종료

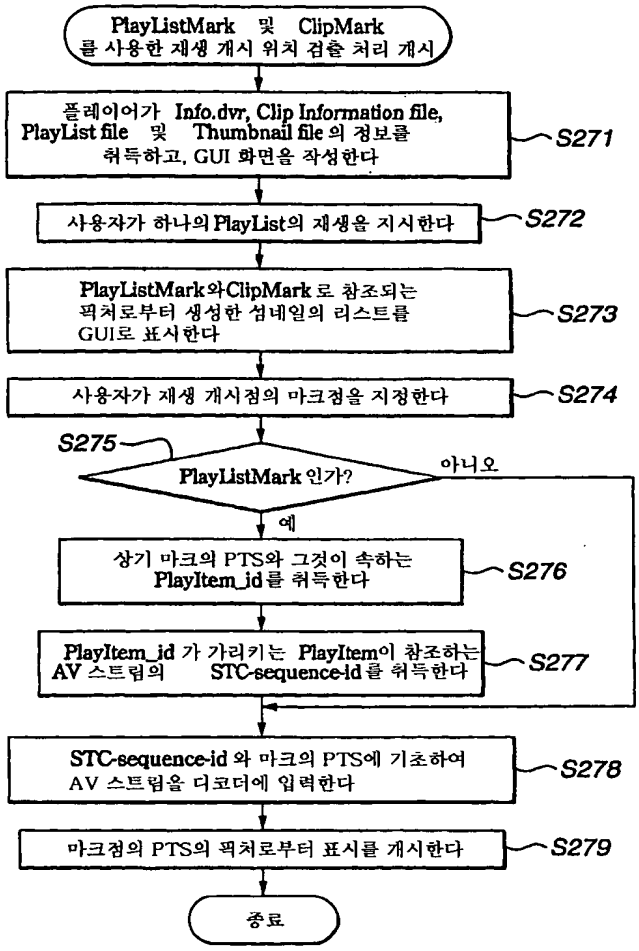
도면132


개시
플레이어가 Info.dvr, Clip Information file, PlayList file 및 Thumbnail file 의 정보를 취득하고, GUI 화면을 작성한다
S241
사용자가 하나의 PlayList 의 재생을 지시한다
S242
현재의 PlayItem의 STC-sequence-id와 IN_time의 PTS 로부터 IN_time 보다 시간적으로 이전에 가장 가까운 엔트리 포인트가 있는 소스 패킷 번호를 취득한다
S243
상기 엔트리 포인트가 있는 소스 패킷 번호로부터 AV 스트림의 데이터를 판독하여 디코더로 공급한다
S244
시간적으로 이전의 PlayItem 과의 표시의 접속 처리를 connection_condition 에 따라 행한다
S245
디코더는 IN_time의 PTS 의 픽처로부터 표시를 개시한다
S246
AV 스트림의 디코드를 계속한다
S247
S248
OUT_time의 PTS의 화상인가 ?
아니오
화상을 표시한다
S249
예
S250
최후의 PlayItem 인가?
아니오
예
종료

도면133


PlayListMark의 작성 개시
플레이어가 Info.dvr, Clip Information file, PlayList file 및 Thumbnail file 의 정보를 취득하고, GUI 화면을 작성한다
S261
사용자가 하나의 PlayList의 재생을 지시한다
S262
상기 지시된 PlayList 의 재생을 개시한다
S263
사용자가 마음에 드는 장면의 부분에 마크의 세트를 지시한다
S264
마크의 PTS와 그것이 속하는 PlayItem 의 PlayItem_id 를 취득한다
S265
마크 정보를 PlayListMark에 스토어한다
S266
PlayList 파일을 기록한다
S267
종료

도면134



도면135

신택스	바이트수	약호
PlayListMark(){		
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
mark_invalid_flag	1	uimsbf
mark_type	7	uimsbf
mark_name_length	8	uimsbf
ref_to_PlayItem_id	16	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_PID	16	uimsbf
ref_to_thumbnail_index	16	uimsbf
mark_name	8*32	bslbf
}		
}		

도면136

값	의미	노트
0x00	Resume-mark	재생 리즘 포인트. PlayListMark()에 있어서 정의되는 재생 리즘 포인트의 수는 0 또는 1 이어야 한다
0x01	Book-mark	PlayList 의 재생 엔트리 포인트. 이 마크는 사용자가 세트할 수 있으며, 예를 들면 마음에 드는 장면의 개시점을 지정하는 마크로 사용한다. 이 마크는 PlayListMark()에 복수있어도 된다
0x02	Chapter-mark	사용자는 PlayList 중에서 하나의 챕터가 이 마크로부터 개시하는 것을 의도하고 있다. 사용자가 세트할 수 있다. 이 마크는 PlayListMark()에 복수있어도 된다
0x03	Skip-start-mark	PlayListMark 중에 하나의 Skip-start-mark 가 세트되는 경우, 이 Skip-start-mark 의 엔트리 직후에 하나의 Skip-end-mark 가 세트되어야 한다. Skip-start-mark 의 타임스탬프로부터 Skip-end-mark 의 타임스탬프까지, 사용자는 PlayList 의 재생을 스킵하는 것을 의도하고 있다 Skip-start-mark와 Skip-end-mark는 동일한 ref_to_PlayItem_id 을 갖는다. 또한, Skip-start-mark와 Skip-end-mark 는 만약 entry_ES_PID가 0xFFFF 가 사용자가 세트할 수 있는 마크로, 이마크는 PlayListMark()에 복수있어도 된다
0x04	Skip-end-mark	
0x05-0x3F	Reserved for future use	Reserved for PlayListMark
0x40-0x7F	Reserved for ClipMark	

도면137

신택스	바이트수	약호
ClipMark(){		
length	32	uimsbf
maker_ID	16	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<*number_of_Clip_marks*; i++){		
mark_invalid_flag	1	uimsbf
mark_type	7	uimsbf
ref_to_STC_id	8	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_PID	16	uimsbf
ref_to_thumbnail_index	16	uimsbf
representative_picture_time_stamp	32	uimsbf
}		
}		

도면138

Mark_type	의미	노트
0x00-0x3F	reserved for future use	Reserved for PlayListMark
0x40	Scene-start-mark	장면의 게시 포인트를 나타내는 마크점
0x41-0x5F	Reserved for common ClipMark	
0x60-0x7F	Maker defined ClipMark	maker_ID 에 의해 나타내는 메이커가 자유롭게 의미를 정의할 수 있다

도면139